LEARNING NONLINEAR EMBEDDINGS FOR CONTROL OF NEURAL DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Control of neural activity could provide a way to study causal interactions within neural populations and develop adaptive interventions for neurological disorders. However, doing so is hindered by the complexity of neural population dynamics and their response to external inputs. Prior work on neural control has largely relied on linear or generalized linear models, whose simplicity enables tractable estimation and control design, but which may not capture sufficient complexity in neural dynamics for accurate control. Here, we develop a model that learns nonlinear embeddings of neural population activity with linear or locally linear latent dynamics using control-focused objectives. Combined with recursive state estimation, this model captures nonlinearity while enabling tractable model-based control. We validate our model and controller in simulation, across benchmark systems relevant to neuroscience, spanning both linear and nonlinear dynamics. Across these systems, our models outperform the baselines in stabilizing activity at desired target setpoints and in restoring movement trajectories following simulated lesions. Together, these simulation results show that learning control-relevant embeddings and latent dynamics provides a practical approach toward future systems for model-based control of nonlinear neural population activity.

1 INTRODUCTION

Enabling control of neural population dynamics is essential both for testing causal hypotheses about neural computation (Peixoto et al., 2021; O'Shea et al., 2022; Vinograd et al., 2024) and for developing adaptive therapies, *e.g.*, with stimulation inputs (Guidotti et al., 2025; Koeglsperger et al., 2026). For both goals, models that predict how stimulation inputs change neural activity could guide the choice of inputs within a controller (Oganesian & Shanechi, 2024). The nature of neural population states, their measurements, and stimulation imposes four requirements on such a model and controller. First, neural observations are high-dimensional but often occupy low-dimensional nonlinear manifolds (Gallego et al., 2017; Duncker & Sahani, 2021; Chaudhuri et al., 2019; Sani et al., 2021; Gardner et al., 2022). Thus, learning dynamics directly in observation space is data-intensive, motivating compact nonlinear representations that retain the neural dynamics relevant to control. Second, the dynamical state of the population is not directly observed: measurements are noisy and partial, and can drop out (Simeral et al., 2021), so the state must be estimated online, with each new observation and input updating the current estimate. Third, safety considerations sharply limit exploratory stimulation in neural applications, so models must be identified sample-efficiently from limited open-loop perturbation data, without closed-loop interaction during learning. Finally, feedback controllers must respect hard bounds on inputs, admit new feasible neural or behavioral targets without retraining the model, and run in real time.

One motivation for developing such a model and controller is deep brain stimulation (Lozano et al., 2019), which illustrates the gap. Current systems commonly apply a fixed pre-programmed input waveform continuously in open loop (Koeglsperger et al., 2026), or trigger or scale stimulation using predefined biomarkers (Little et al., 2013; Velisar et al., 2019; Bocci et al., 2021; Oehrn et al., 2024), with no learned model to optimize stimulation based on the predicted evolution of the neural state. Learning such a model is the aim of a large literature on latent state-space modeling of neural population dynamics, with nonlinear observation models and linear (Gao et al., 2016), locally linear (Fraccaro et al., 2017; Karl et al., 2017), or nonlinear (Chung et al., 2015; Krishnan et al., 2015; Pandarinath et al., 2018; Sani et al., 2024) latent dynamics. However, most such models are designed

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

for offline analysis of recorded activity rather than for control. They typically do not model the effect of external inputs on the dynamics, and they use noncausal or windowed inference rather than recursive online state estimation (Gao et al., 2016; Pandarinath et al., 2018), precluding real-time control in their published form. Sequential variational state-space models (Fraccaro et al., 2017; Karl et al., 2017) provide inference causally over time, but likewise do not address neural controller design. Recent work goes further for neural data, combining nonlinear embeddings of neural observations with linear-Gaussian latent dynamics, enabling Kalman filtering even when observations are missing (Abbaspourazad et al., 2024), with extensions to multiscale neural activity (Erturk & Shanechi, 2025). However, these models were developed for prediction rather than control, and predictive accuracy is not guaranteed to identify the model that gives the best control performance (Lambert et al., 2020).

Developing control-focused dynamical models has proven difficult. Closed-loop continuous control using a dynamical model has remained elusive at the level of neural populations *in vivo*, with *in vivo* demonstrations limited to the firing rate of single neurons using linear models (Bolus et al., 2021). Other *in vivo* dynamical modeling studies have learned linear or generalized linear state-space models with input but without control, including models learned from responses to electrical stimulation (Yang et al., 2021) or sensory stimuli (Vahidi et al., 2024). In computational simulation studies, linear models have been paired with linear-quadratic-Gaussian (Yang et al., 2018), linear-quadratic-integral (Davila et al., 2022), adaptive (Santaniello et al., 2011), static output feedback (Ashourvan et al., 2020), or model-predictive (Haddock et al., 2017; Steffen et al., 2024) control. These linear approaches are efficient for inference and amenable to standard controller design, but restrict the dynamics and observation geometry to be linear, thus not meeting the first requirement.

Outside of neuroscience, learned latent state models partially address this limitation by learning nonlinear embeddings with linear or locally linear input-driven latent dynamics that remain amenable to planning for control (Watter et al., 2015; Morton et al., 2018; Bounou et al., 2021). However, these models were developed for settings where observations are clean and complete renderings of the state; the encoder itself therefore serves as the state estimator, mapping the current observation or a fixed window directly to the latent state. This setting is fundamentally different from neural data, where the state is only partially observed through noisy measurements. In neural applications, in-silico simulation studies have used Koopman dynamics learned from windowed observations (Liang et al., 2023) and linear latent dynamics initialized from a pretrained nonlinear embedding (Fehrman & Meliza, 2025) for control. These formulations inherit the same encoder-as-estimator structure: the state is a deterministic function of recent measurements, with no filtering to reduce noise in neural measurements or bridge missing neural samples, not meeting the second requirement.

Finally, reinforcement learning (RL) offers direct policy optimization. However, model-free RL is sample inefficient, requiring extensive stimulation data that is typically not feasible to collect from the brain. Furthermore, both model-free and model-based online RL requires extensive closed-loop interaction with the plant (Nagabandi et al., 2018; Hafner et al., 2019), and safety considerations preclude such extensive exploratory stimulation in the brain. As such, online RL does not meet the third requirement. Moreover, both online and offline RL produce policies tied to the reward and target representation specified during training, not meeting the fourth requirement. Thus, RL for neural control has largely been confined to simulated systems (Mitchell & Petzold, 2018; Pan et al., 2024), and *in-vivo* uses of RL do not control population activity: they learn stimulation policies for the firing of single neurons (Coventry et al., 2024), for behavioral objectives in *C. elegans* (Li et al., 2024), or, with offline RL, for changing stimulation based on a predefined scalar biomarker (Gao et al., 2023).

In sum, no prior method meets all four requirements.[1] We introduce ConFINE (**Con**trol-**F**ocused **I**dentification of **N**eural **E**mbeddings), a controlled latent state model that does, in a single framework:

- It captures population activity in a compact latent state that retains the dynamics relevant to control while maintaining nonlinear structure and expressivity, unlike linear models. This is done by training a nonlinear embedding of neural observations jointly with linear or locally linear input-driven latent dynamics. Further, multi-horizon prediction and latent-state regularization shape this state for closed-loop performance rather than predictive fit alone.
- It tracks the latent state in real time from noisy observations and carries the estimate through missing samples, unlike prior latent models for control that encode only current or recent

[1] Appendix A discusses alternative approaches and their limitations, such as mechanistic models or static inverse maps, which are still further removed from our approach.

observations. This is done by learning stochastic linear or locally linear latent dynamics that allow the state to be tracked by Kalman filtering (Anderson & Moore, 1979), incorporating each observation and input as it arrives.

- It enables model-based control, with the strong inductive bias of a linear or locally linear latent backbone, allowing sample-efficient system identification from finite open-loop perturbation data.
- It admits new feasible neural or behavioral targets without retraining the model and computes bounded inputs in real time. This is achieved as the same linear structure admits classical real-time controllers such as linear-quadratic-Gaussian (Stengel, 2012) and model predictive control (Rawlings et al., 2017); neural targets enter only through the controller's cost, and behavioral targets enter through behavioral readouts of the latent state, trained jointly or fit post hoc.

We evaluate ConFINE on a wide range of simulated plants, including linear and nonlinear dynamics and a circuit model fit to movement kinematics, and across these the control-focused variants consistently achieve the strongest closed-loop performance among the learned baselines.

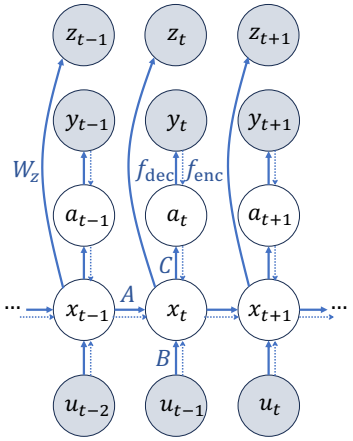

Figure 1: ConFINE graphical model. Solid lines indicate generative model, dashed indicate inference. Shaded indicates observed, white indicates latent variables.

2 METHODS

2.1 MODEL ARCHITECTURE

To meet these requirements, we propose a model with two components: an input-driven (locally) linear latent dynamical system that models the temporal evolution of the state, and a neural-network embedding that captures the nonlinear manifold on which the observed activity resides (Fig. 1). The embedding absorbs the nonlinearity of the observation geometry so that the latent dynamics remain (locally) linear, which is what makes recursive state estimation and constrained planning tractable; control-motivated training then shapes these dynamics for control rather than predictive fit alone, which is the focus of prior autonomous models without input (Abbaspourazad et al., 2024). Given neural observations y_t (*e.g.*, firing rates, field potentials, calcium imaging traces, or other modalities) driven by inputs u_t (*e.g.*, electrical or optogenetic stimulation), we model their dynamics:

$$x_{t+1} = A_{t|t}\, x_t + B_{t|t}\, u_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, \Sigma_{x,t|t}) \tag{1}$$

$$a_t = C_{t|t-1}\, x_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \Sigma_{a,t|t-1}) \tag{2}$$

$$y_t = f_{\text{dec}}(a_t) \tag{3}$$

where x_t is the latent dynamical state, a_t is a latent representation of neural activity, f_{dec} is a nonlinear decoder implemented by a neural network, and η_t and ε_t are noises. When behavioral data z_t is available (*e.g.*, cursor position or muscle activation), we also model behavior as an affine function of the latent to enable efficient controller design using linear methods (§2.4):

$$z_t = W_{z,t|t-1}\, x_t + b_{z,t|t-1} \tag{4}$$

The state-dependent matrices $M_{s|t} := M(\hat{x}_{s|t})$ for $M \in \{A, B, C, \Sigma_x, \Sigma_a, W_z, b_z\}$ are generated by a neural network as a function of $\hat{x}_{s|t}$, the estimate of x_s given observations through time t (see §2.2), thus making the model locally linear. We also consider a linear variant where these parameters $M_{s|t} := M$ are not state-dependent and are instead learned globally (see also App. E).

2.2 INFERENCE

To estimate the latent state (x_t, a_t) from observed neural activity $y_{1:t}$, we first define an encoder f_{enc}, implemented, like the decoder, as a neural network, to nonlinearly map the observation to the latent space, $\tilde{a}_t = f_{\text{enc}}(y_t)$. Since the latent dynamics are (locally) linear by construction, we can now

perform exact inference in the presence of noise using Kalman filtering (Anderson & Moore, 1979) to recursively estimate the dynamical latent state x_t, treating $\tilde{a}_t$ as the noisy observation. The estimate is $\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(\tilde{a}_t - C_{t|t-1}\hat{x}_{t|t-1})$, where K_t is the Kalman gain (see App. E.1).

2.3 Prediction and training

Safety and ethical considerations can make closed-loop exploration impractical in neural stimulation applications. Instead, for training, data sequences $\{u_{1:T}, y_{1:T}, z_{1:T}\}$ are collected in open-loop by stimulating the plant (*i.e.*, the system to be controlled) with a bounded persistently exciting input u_t (Yang et al., 2018; 2021; Bolus et al., 2021) and measuring the resulting neural activity y_t and, if applicable, behavior z_t. We thus use no closed-loop interaction with the plant during learning.

Because the controller optimizes inputs over open-loop rollouts of the learned model, the error that matters is the model's accuracy over the planning horizon rather than at a single step. We therefore train on k-step-ahead prediction: starting with the filtered state $\hat{x}_{t|t}$, we use the next k inputs $u_{t:t+k-1}$ to recursively propagate the dynamics forward for k steps, using the transition function (1) to predict the latent state $\hat{x}_{t+k|t}$. We then predict neural activity $\hat{y}_{t+k|t}$ through the observation matrix (2) and the decoder (3); see App. E.1. The encoder/decoder, as well as the dynamic model parameters are then learned end-to-end to minimize multi-step prediction mean-squared error (MSE):

$$\mathcal{L}_y = \sum_{k \in \mathcal{K}_y} \lambda_k^y \sum_{t=1}^{T-k} ||\hat{y}_{t+k|t} - y_{t+k}||^2 \tag{5}$$

When control of behavior is of interest, accurate prediction of neural observations alone may not yield a latent representation that adequately predicts the behavioral consequences of stimulation (see App. D.3). If available, paired behavioral data z_t can be used to supervise training: we jointly train the model to predict behavior $\hat{z}_{t+k|t}$ from the filtered ($k = 0$) and predicted ($k > 0$) states $\hat{x}_{t+k|t}$ through the behavioral readout (4), minimizing the MSE between predictions and data:

$$\mathcal{L}_z = \sum_{k \in \mathcal{K}_z} \lambda_k^z \sum_{t=1}^{T-k} ||\hat{z}_{t+k|t} - z_{t+k}||^2 \tag{6}$$

Behavior thus serves only as a supervisory training signal; online state estimation uses only neural observations and inputs. As in prior models of behaviorally relevant neural dynamics (Sani et al., 2021; 2024), supervision determines which neural dynamics the latent state prioritizes. This mirrors the setting of many neural interfaces, where behavioral or symptom measurements are available during calibration but not continuously during deployment.

Finally, because the controller plans by rolling the learned dynamics forward without observations, we encourage models whose dynamics carry the state on their own rather than relying on correction by each new observation. To do so, we penalize the KL divergence D_t between the filtered posterior and the one-step predictive prior, which measures the change in the latent-state distribution induced by the current observation. A free-nats threshold α removes KL penalty gradients for small belief updates (Kingma et al., 2016; Hafner et al., 2019):

$$D_t = D_{\mathrm{KL}}\big(\mathcal{N}(\hat{x}_{t|t},\, P_{t|t}) \,\big\|\, \mathcal{N}(\hat{x}_{t|t-1},\, P_{t|t-1})\big)\,, \qquad \mathcal{L}_{\mathrm{KL}} = \sum_{t=1}^{T} \max(D_t,\, \alpha)\,. \tag{7}$$

Our KL penalization is in the spirit of prior variational objectives (Chung et al., 2015; Krishnan et al., 2015; Hafner et al., 2019) or consistency penalties (Watter et al., 2015). The total loss is then: $\mathcal{L} = \mathcal{L}_y + \mathcal{L}_z + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}}$. Further details are described in App. E.2.

2.4 Control

Once identified, the model supports real-time control without further learning: the same model serves any feasible target, and targets, costs, and input bounds enter only the optimization solved at each step. At each step, a receding-horizon model predictive controller (MPC) plans box-constrained inputs $u_{t:t+H-1}$ from the state estimate $\hat{x}_{t|t}$ (Mayne et al., 2000; Muske & Rawlings, 1993), with targets specified as either desired neural observations or behavioral outputs.

One challenge is that a target observation y^* does not by itself identify a dynamically feasible latent-state target. Its encoding $\tilde{a}^* = f_{\text{enc}}(y^*)$ specifies a desired latent observation, but multiple dynamical states may produce the same latent observation, and such a state need not be maintainable under the learned dynamics. For the globally linear model used in our observation-target experiments, we therefore jointly select a latent target x^* and steady input u^*, similar to the steady-state target calculation of Muske & Rawlings (1993):

$$(x^*, u^*) = \arg\min_{x,u} \; ||Cx - \tilde{a}^*||^2 + \lambda_{\text{eq}}||Ax + Bu - x||^2 \quad \text{s.t.} \quad u_{\min} \le u \le u_{\max}, \tag{8}$$

The first term encourages the selected state to reproduce the target observation, while the second, weighted by λ_{eq}, encourages (x^*, u^*) to form an equilibrium of the learned dynamics (App. G). The MPC then regulates the predicted state and input toward (x^*, u^*) using quadratic state and input costs (Muske & Rawlings, 1993).

When behavior z_t is available, the controller can instead optimize the difference between the predicted behavior $\hat{z}_{t+k|t}$ and the desired behavior along the planning horizon without constructing a latent-state target. In the behaviorally supervised models, the affine readout (4) is learned jointly with the dynamics through $\mathcal{L}_z$. For models trained without behavioral supervision, we fit the readout after training on the model's own state estimates:

$$(W_z, b_z) = \arg\min_{W,b} \sum_{k \in \mathcal{K}_z} \sum_{t=1}^{T-k} ||W\hat{x}_{t+k|t} + b - z_{t+k}||^2, \tag{9}$$

Including positive prediction horizons in $\mathcal{K}_z$ (Tab. 4) fits the readout on the future-predicted states it uses during planning, rather than only on filtered estimates.

Another challenge is that for the locally linear model, the dynamics matrices depend on the predicted trajectory and thus on the planned inputs, so, unlike the globally linear model, the planning problem is no longer a quadratic program. We thus recover a sequence of quadratic programs by iteratively rolling out a nominal input sequence, evaluating the state-dependent matrices along the resulting trajectory, and re-optimizing the inputs with those matrices held fixed, similar to the quasi-LPV MPC iteration of Cisneros et al. (2016). Details of optimization problems and controllers are in App. F.

3 Results

3.1 Linear dynamics on nonlinear manifolds

We first evaluate ConFINE on synthetic plants whose latent dynamics are linear but whose neural observations y_t live on a nonlinear manifold:

$$x_{t+1} = Ax_t + u_t + q_t, \qquad q_t \sim \mathcal{N}(0, \Sigma_q) \tag{10}$$

$$y_t = m(x_t) + s_t, \qquad s_t \sim \mathcal{N}(0, \Sigma_s) \tag{11}$$

where $m(\cdot)$ is a nonlinear manifold embedding – either ring-like (Chaudhuri et al., 2019), toroidal (Gardner et al., 2022) or Swiss roll manifold (Tenenbaum et al., 2000). This setting isolates the model's representational and control capacity from the additional challenge of approximating nonlinear dynamics, since the ground-truth latent dynamics are exactly representable by the model's linear backbone. We do not define a behavioral output, so training is done without behavioral supervision.

For each of these manifolds, we vary the plant's dynamics parameters (*i.e.*, the matrix A) to produce families of plants with specific dynamical structure, covering a large range of biologically relevant scales and phenomena. For the ring manifold – defined by a one-dimensional latent space – we consider a family of plants whose latent dynamics decay at varying timescales. For the torus and Swiss roll manifolds – defined by two latent dimensions, enabling a much richer set of dynamical regimes – we consider three families, each parameterized by two values: (i) two-timescale decay with independently varying timescales (ii) rotations with varying decay and oscillation frequencies (iii) non-normal dynamics with varying timescales and non-normality metrics. All values are chosen to match biologically relevant ranges from the literature – see App. G.1 for detailed biological rationale and definitions of the plant families.

In Tab. 1 (see also Fig. A3), we report control performance of our model, ablations, and baselines, averaged across multiple plants from each of these seven families. For each family, we sweep over

Table 1: Control performance on the manifold plants: distance to target averaged over the last 10 steps of each episode, mean ± s.e. over each family's plants, with oracle and chance levels for reference. Lower is better; **best**, second best. Details in App. G.1.

	plant →	Ring	Torus			Swiss roll		
model ↓		scalar	rotation	decay	shear	rotation	decay	shear
ConFINE	KL	**0.24 ± 0.01**	**0.78 ± 0.03**	**0.55 ± 0.05**	1.18 ± 0.10	**0.29 ± 0.01**	**0.76 ± 0.05**	1.77 ± 0.14
	no KL	**0.24 ± 0.01**	0.80 ± 0.03	0.61 ± 0.06	**1.08 ± 0.10**	**0.29 ± 0.01**	**0.76 ± 0.05**	**1.53 ± 0.12**
AE+LDM		0.80 ± 0.08	2.93 ± 0.04	1.62 ± 0.07	2.98 ± 0.08	1.23 ± 0.04	0.84 ± 0.03	7.43 ± 2.20†
LDM		1.24 ± 0.02	3.85 ± 0.08	3.74 ± 0.02	4.15 ± 0.04	1.41 ± 0.03	2.07 ± 0.11	23.40 ± 4.23†
Oracle		0.17 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.18 ± 0.00	0.18 ± 0.00	0.18 ± 0.00
Chance		1.69	5.66	5.66	5.66	3.40	3.40	3.40

† Worse than chance: the Swiss roll is unbounded, so a trajectory can end arbitrarily far from the target.

the relevant range of its parameters to generate instances of the plant. For example, Fig. 2 (left) shows the parameter sweep over the rotational family (τ, θ) on the toroidal manifold, with each point on the heatmap representing the average control error on that plant (others in Figs. A1 and A2).

Specifically, we perform open-loop system identification and evaluate performance on subsequent control tasks. We choose control targets x^* on a uniform grid over the manifold's latent space (Fig. 2, middle), convert them to observation targets $y^* = m(x^*)$, run closed-loop control to generate controlled trajectories for each target (Fig. 2, right), and report the error as the distance between the target and the measured observation at the tail of the control episode (see App. G.1 for details). In each case, using the same MPC throughout, ConFINE outperforms two baselines: a linear dynamical model (LDM), and a model with the same two components as ConFINE– an autoencoder and an LDM – but trained separately (AE+LDM: an autoencoder pretrained with gradient descent, followed by an LDM fit to its embeddings); App. E.2 describes how the baselines are fit.

These comparisons also show the benefit of the nonlinear embedding and of joint training of model components. Given the low-dimensional structure of the latent space, we can visualize both the latent space and the observation manifold each model learns and compare them with the ground truth (Fig. 3). ConFINE reliably learns a latent space that is approximately isomorphic to that of the ground truth,[2] and a decoder whose image reproduces the toroidal geometry of the true manifold. Neither baseline recovers this structure, and both miss the target: LDM's identity encoder can only represent the manifold as the linear subspace spanned by its observation matrix, while AE+LDM recovers a torus-like manifold but learns a malformed latent. Nonlinearity in the embedding is therefore not by itself sufficient; the embedding and the dynamics must be trained *jointly*. Additional plants are shown in Figs. A4 to A6.

[2] Even in minimal linear state-space models, latent coordinates are only identifiable up to an invertible linear map (Kailath, 1980).

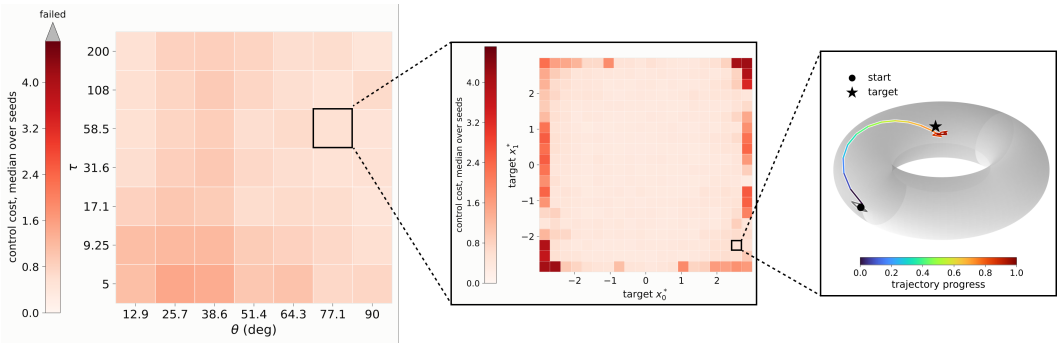

Figure 2: Control on the torus rotation family with ConFINE. Left: control performance of every plant in the (τ, θ) sweep, median over seeds. Middle: control performance for each of the (20×20) targets of one plant, median over seeds. Right: an example trajectory on the torus for one target.

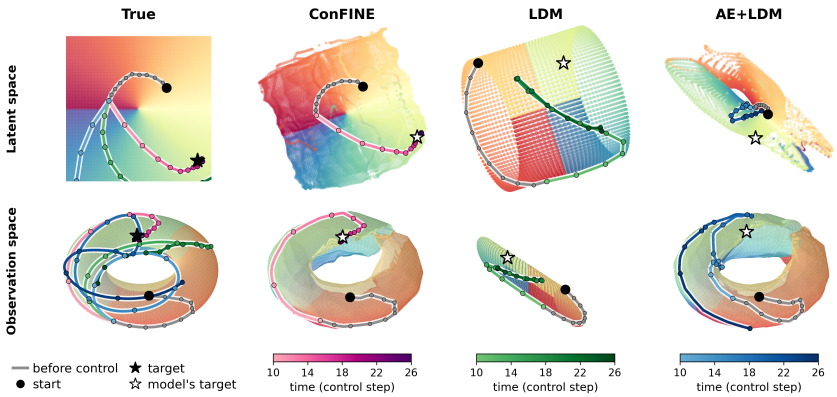

Figure 3: Learned latent spaces (top) and manifolds (bottom) on a torus rotation plant. First column: the ground truth, with the plant trajectory under each model's controller (ConFINE pink, LDM green, AE+LDM blue; steps before control onset shown in gray). Remaining columns: each model's learned latent space and manifold, with its trajectory and its own estimate of where the target lies (hollow star).

3.2 Nonlinear dynamics in a recurrent neural network

We next consider a more general class of plants – task-trained recurrent neural networks (RNNs). Such plants provide a general framework for modeling neural computation (Yang et al., 2019; Driscoll et al., 2024), including motor applications such as control of biomechanical arms (Miconi, 2017; Kalidindi et al., 2021) and reproducing recorded muscle activity (Sussillo et al., 2015). Following models trained on simulated reaching trajectories (Michaels et al., 2016; Feulner & Clopath, 2021), we train an RNN to generate center-out trajectories and use it as a nonlinear plant of the form:

$$h_{t+1} = \phi(W_h h_t + W_u u_t) + q_t, \qquad q_t \sim \mathcal{N}(0, \Sigma_h) \tag{12}$$

$$y_t = g_\theta(h_t) + r_t, \qquad r_t \sim \mathcal{N}(0, \Sigma_y) \tag{13}$$

$$z_t = W_z h_t \tag{14}$$

with a nonlinear observation map g_θ and a linear behavioral readout W_z, paralleling latent dynamical models that combine nonlinear mappings to neural activity with linear decoding of movement from latent factors (Pandarinath et al., 2018).

We consider three variants of this plant with different amounts of process noise and ranges of initial states (see App. G.2). As before, models are identified from the network's responses to random inputs, observing the neural activity y_t but not the behavior, whose readout (9) is fit after training. The MPC (§2.4) must then drive z_t from a random starting position to one of several fixed targets and hold it there. We compare ConFINE, with and without the KL term, to LDM, AE+LDM, and an oracle MPC with access to the true plant and state. The cost is the squared distance to the target in behavior space z, rather than in observation space y as for the manifolds, plus an input penalty weighted by R.

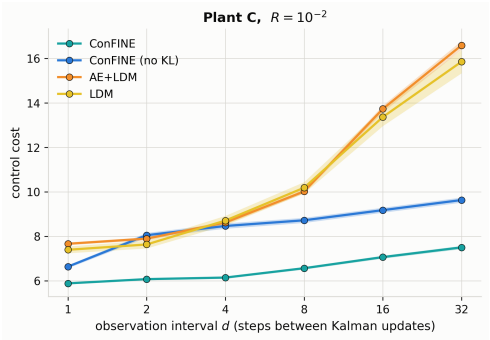

Figure 4: Control under intermittent observation on RNN plant C ($R = 10^{-2}$): control cost against the observation interval d, with the Kalman update applied every d steps while the plant and controller run at every step.

ConFINE with the KL term achieves the lowest cost for every plant and R (Tab. 2) ; without KL, ConFINE, AE+LDM, and LDM perform similarly on plant A and separate only with more spread-out initial states (plant B) or larger process noise (plant C). The KL term also substantially reduces the variability of the cost across training seeds (Fig. A8), and in an example episode its reaches end closer to the targets, whereas those of LDM and AE+LDM overshoot and drift (Fig. A7). Reducing the input penalty R allows the controller to apply larger inputs, which lowers the oracle's cost on every plant; on plant A, however, it raises the cost of every model except ConFINE with KL, suggesting that more aggressive control amplifies their errors.

Table 2: Control performance on the reach-trained RNN plants: closed-loop cost, mean ± s.e., with oracle and do-nothing (zero input) costs for reference. Lower is better; **best**, second best. Details in App. G.2.

	plant →	RNN "A"		RNN "B"		RNN "C"	
model ↓		$R = 10^{-2}$	$R = 10^{-3}$	$R = 10^{-2}$	$R = 10^{-3}$	$R = 10^{-2}$	$R = 10^{-3}$
ConFINE	KL	**1.82 ± 0.01**	**1.79 ± 0.01**	**2.09 ± 0.02**	**1.75 ± 0.02**	**3.57 ± 0.03**	**3.41 ± 0.04**
	no KL	2.21 ± 0.02	2.32 ± 0.03	2.22 ± 0.03	1.99 ± 0.02	3.83 ± 0.04	3.71 ± 0.04
AE+LDM		2.14 ± 0.02	2.36 ± 0.02	2.66 ± 0.03	2.58 ± 0.03	4.40 ± 0.04	4.24 ± 0.04
LDM		2.17 ± 0.07	2.40 ± 0.08	2.53 ± 0.09	2.46 ± 0.09	4.22 ± 0.11	4.07 ± 0.12
Oracle		1.42 ± 0.01	1.16 ± 0.02	1.54 ± 0.03	1.27 ± 0.03	2.25 ± 0.06	1.87 ± 0.05
Do nothing		19.4	19.4	19.6	19.6	21.9	21.9

Finally, the controller plans using only the learned dynamics, without observations; dropped samples (Simeral et al., 2021) impose the same demand on the estimator, and make this property directly measurable. To evaluate this, we apply the Kalman update only every d steps during control (plant C shown in Fig. 4; remainder in Fig. A9). By $d = 32$, the cost of LDM and AE+LDM grows 2-3×, and that of ConFINE without the KL term grows by 20–40% already at $d = 2$, whereas ConFINE with the KL term degrades by only ∼10% or less up to $d = 8$ and only ∼30% at $d = 32$: between observations its belief is propagated by the learned dynamics, which the KL term encourages to require only small corrections from each new observation (§2.3).

3.3 Simulated stroke recovery in a model of grasping

Finally, moving toward a more biological plant, we use the pretrained cortical network model of Michaels et al. (2020), trained on recorded grasping kinematics and evaluated against neural recordings, in the setting developed by Bryan et al. (2023), using stimulation to restore movement following simulated lesions to the circuit (see also Pan et al., 2024; Dura-Bernal et al., 2016). The plant is an RNN consisting of three modules – representing brain areas AIP, F5, and M1 – whose M1 output drives the kinematics of the non-human primate's reach-to-grasp movement (Fig. 5). Following Bryan et al. (2023), we observe the circuit through random projections of its activity, stimulate it through bounded input channels, and simulate four stroke-like lesions that silence units within an area, cut the projections between areas, or both (see App. G.3).

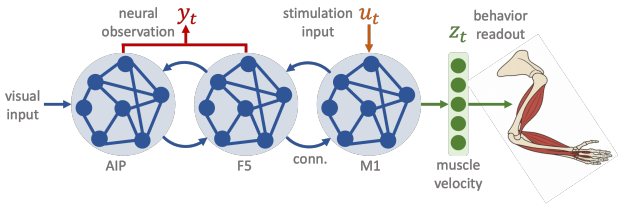

Figure 5: Schematic of the plant from Michaels et al. (2020) fit to primate kinematic data in an object grasping task. Neurons in AIP, M1, and/or connections are lesioned and our controller stimulates M1 to restore healthy movement.

As before, models are identified only from the responses of the lesioned circuit to random open-loop stimulation. Since behavior is available here, we compare two protocols: *unsupervised*, in which the behavioral readout is fit after training via (9), and *supervised*, in which it is learned jointly with the dynamics (§2.3). As on the RNN plant (§3.2), the controller targets behavior rather than neural activity, but unlike both previous settings, the task is *tracking* rather than regulation: instead of driving the system to a fixed target, the controller must follow a time-varying reference, given on each trial by the animal's recorded kinematics (Michaels et al., 2020).

We compare the globally and locally linear variants of ConFINE (with and without the KL term) with LDM and AE+LDM, all identified from the same offline dataset under the same protocol. Results from Bryan et al. (2023) are included for reference.[3] We quantify performance as percent recovery, summarized in Tab. 3 and detailed in Figs. A10 to A13.

[3]Comparison to Bryan et al. (2023) is not like-for-like and only included as a reference. The authors train in closed loop on a behavioral task loss, refitting a circuit emulator on data from the policy's own stimulation, producing a task-specific stimulation policy which must be retrained for a new task, not a dynamical model that a controller can use to track an arbitrary reference trajectory.

Table 3: Simulated stroke recovery in the grasping-circuit model: percent recovery of the behavioral loss, mean ± s.e., with the readout fit post hoc (Unsup.) or jointly (Sup.). See App. G.3 for interpretation of negative values and values over 100. Higher is better; **best**, second best. (LL: Locally Linear).

	lesion →	M1		AIP		Conn.		AIP+F5+conn.	
model ↓		Unsup.	Sup.	Unsup.	Sup.	Unsup.	Sup.	Unsup.	Sup.
ConFINE	KL	11.2±1.1	70.9±0.8	-326.9±7.8	73.3±1.0	73.2±0.5	94.5±0.3	35.1±2.0	74.1±1.1
	no KL	6.4±1.0	68.9±0.8	-274.6±8.6	54.7±1.4	80.3±0.5	93.9±0.3	**53.9±1.7**	73.7±1.1
LL-ConFINE	KL	22.3±1.5	83.6±0.5	**-52.1±3.6**	**100.6±0.9**	82.9±0.5	98.9±0.2	33.5±1.8	**76.1±1.1**
	no KL	**28.3±1.0**	**85.6±0.4**	-157.7±7.5	97.4±1.0	**84.2±0.4**	**99.1±0.2**	46.9±1.7	73.4±1.1
AE+LDM		5.9±1.3	58.9±0.6	-360.9±9.0	-149.2±4.5	-1.3±2.5	79.4±0.4	16.1±2.3	10.8±2.4
LDM		18.2±1.5	61.6±0.8	-185.8±5.7	-115.5±3.9	62.6±0.7	86.3±0.3	46.5±1.7	52.7±1.4
Bryan et al. (2023)[3]		75.5		13.9		90.5		–	

Behavioral supervision is critical on this plant, improving recovery for every model and lesion except one. Unlike the RNN plant, the behavior is a readout of one area of a three-area circuit observed through a lossy projection, so a latent state that predicts the neural observations need not retain what is needed to predict movement; accordingly, recovery correlates strongly with behavioral but not neural prediction error across models (Fig. A14). With supervision, every variant of ConFINE outperforms both baselines on every lesion, and the locally linear (LL) variants achieve the highest recovery on the M1, AIP, and connection lesions, approaching healthy performance on the latter two. This includes the most challenging lesion, AIP, where no model recovers behavior without supervision and both baselines do worse than no stimulation.

4 Discussion

We introduce ConFINE, a controlled latent state-space model that jointly learns a nonlinear embedding of neural population activity and (locally) linear latent dynamics, trained with control-oriented objectives. These components address the requirements of model-based neural control: the learned embedding maps activity on nonlinear manifolds to a compact latent state that retains control-relevant dynamics; a Kalman filter estimates this state recursively, bridging noisy or missing measurements; the model is identified entirely from bounded open-loop perturbation data; and the linear backbone admits real-time classical MPC, which handles input bounds and new targets without retraining.

On linear dynamics over nonlinear manifolds (§3.1), ConFINE outperforms the baselines across a wide range of biologically relevant dynamical regimes and reliably learns a meaningful latent space. In a task-trained RNN (§3.2), the KL term consistently improves control, reduces variability across training seeds, and, critically, makes control robust to sample dropout. In simulated stroke recovery (§3.3), behavioral supervision meaningfully improves control as neural observations are not limited to behaviorally relevant dynamics, and the locally linear variant, which captures nonlinear dynamics while preserving tractable estimation and control, achieves the highest recovery, approaching healthy performance on two of the four lesions. These results show that learning control-relevant embeddings and latent dynamics is a practical approach to model-based control of nonlinear population activity.

Limitations and future work. Our training objectives are target-agnostic proxies for control performance, and without behavioral supervision the latent state need not retain the dynamics relevant to control (§3.3). Tighter coupling to control needs closed-loop data (Gopaluni et al., 2003; Hjalmarsson, 2005) or target-specific objectives (Lambert et al., 2020), violating two requirements for neural control. Future work could design the open-loop excitation for control relevance, or, where safety permits, refine the model under its own controller to track nonstationarity such as plasticity and electrode drift. Our controllers are also certainty-equivalent, using only the mean of the Kalman belief; the covariance could support chance constraints and uncertainty-aware planning (Mesbah, 2016). Similar to many prior approaches, the controller has no formal closed-loop guarantees; data-driven bounds on model error could enable robust output-feedback MPC (Rawlings et al., 2017). Most fundamentally, all results in this work are in simulation; closed-loop continuous control of neural population activity *in vivo* remains elusive, and achieving it is a critical next step for the field.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and to generate synthetic data sets, by interfacing with our computing cluster to launch and manage the jobs that generated the data sets and trained the models. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean and reformat datasets, or interpret results, and assisting with translation and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create and edit software code (including writing, debugging, and refactoring), create scientific figures, compile results tables, draft parts of the paper, edit the paper for readability and correctness (including finding and fixing errors and placing figures), and search for and identify relevant literature. We have reviewed all AI-assisted work: all AI-assisted code was manually read and reviewed by the authors, all AI-assisted text and figures were reviewed and revised by the authors, and all AI-identified references were checked against the original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Hamidreza Abbaspourazad, Eray Erturk, Bijan Pesaran, and Maryam M Shanechi. Dynamical flexible inference of nonlinear latent factors and structures in neural population activity. *Nature Biomedical Engineering*, 8(1):85–108, 2024. doi: 10.1038/s41551-023-01106-1. URL https://doi.org/10.1038/s41551-023-01106-1.

Brian D. O. Anderson and John B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

Arian Ashourvan, Sérgio Pequito, Ankit N. Khambhati, Fadi Mikhail, Steven N. Baldassano, Kathryn A. Davis, Timothy H. Lucas, Jean M. Vettel, Brian Litt, George J. Pappas, and Danielle S. Bassett. Model-based design for seizure control by stimulation. *Journal of Neural Engineering*, 17(2):026009, 2020. doi: 10.1088/1741-2552/ab7a4e. URL https://doi.org/10.1088/1741-2552/ab7a4e.

Malbor Asllani, Renaud Lambiotte, and Timoteo Carletti. Structure and dynamical behavior of non-normal networks. *Science Advances*, 4(12):eaau9403, 2018. doi: 10.1126/sciadv.aau9403.

Giacomo Barzon, Anandita De, Isaac Moran, Conner Carnahan, Luca Mazzucato, and Roozbeh Kiani. Control of cortical population activity with patterned microstimulation. *bioRxiv*, pp. 2026.03.02.709018, 2026. doi: 10.64898/2026.03.02.709018. URL https://doi.org/10.64898/2026.03.02.709018. Preprint.

Tommaso Bocci, Marco Prenassi, Mattia Arlotti, Filippo Maria Cogiamanian, Linda Borellini, Elena Moro, Andres M. Lozano, Jens Volkmann, Sergio Barbieri, Alberto Priori, and Sara Marceglia. Eight-hours conventional versus adaptive deep brain stimulation of the subthalamic nucleus in Parkinson's disease. *npj Parkinson's Disease*, 7(1):88, 2021. doi: 10.1038/s41531-021-00229-z. URL https://doi.org/10.1038/s41531-021-00229-z.

M. F. Bolus, A. A. Willats, C. J. Rozell, and G. B. Stanley. State-space optimal feedback control of optogenetically driven neural activity. *Journal of Neural Engineering*, 18(3):036006, March 2021. ISSN 1741-2552. doi: 10.1088/1741-2552/abb89c. URL https://dx.doi.org/10.1088/1741-2552/abb89c.

Giulio Bondanelli, Thomas Deneux, Brice Bathellier, and Srdjan Ostojic. Network dynamics underlying OFF responses in the auditory cortex. *eLife*, 10:e53151, March 2021. ISSN 2050-084X. doi: 10.7554/eLife.53151. URL https://doi.org/10.7554/eLife.53151.

Oumayma Bounou, Jean Ponce, and Justin Carpentier. Online Learning and Control of Complex Dynamical Systems from Sensory Input. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27852–27864. Curran Associates, Inc., November 2021. URL https://openreview.net/forum?id=x6tV8QhHjs1.

Anatol Bragin, Jerome Engel, Jr., Charles L. Wilson, Itzhak Fried, and György Buzsáki. High-frequency oscillations in human brain. *Hippocampus*, 9(2):137–142, 1999. doi: 10.1002/(SICI)1098-1063(1999)9:2<137::AID-HIPO5>3.0.CO;2-0.

Anatol Bragin, Jerome Engel, Jr., and Richard J. Staba. High-frequency oscillations in epileptic brain. *Current Opinion in Neurology*, 23(2):151–156, 2010. doi: 10.1097/WCO.0b013e3283373ac8.

Matthew J. Bryan, Linxing Preston Jiang, and Rajesh P. N. Rao. Neural co-processors for restoring brain function: results from a cortical model of grasping. *Journal of Neural Engineering*, 20 (3):036004, May 2023. ISSN 1741-2552. doi: 10.1088/1741-2552/accaa9. URL https://doi.org/10.1088/1741-2552/accaa9.

Matthew J. Bryan, Felix Schwock, Azadeh Yazdan-Shahmorad, and Rajesh P. N. Rao. Temporal basis function models for closed-loop neural stimulation. *Journal of Neural Engineering*, 22(5):056012, 2025. doi: 10.1088/1741-2552/ae036a. URL https://doi.org/10.1088/1741-2552/ae036a.

György Buzsáki and Andreas Draguhn. Neuronal oscillations in cortical networks. *Science*, 304 (5679):1926–1929, 2004. doi: 10.1126/science.1099745.

György Buzsáki, Costas A. Anastassiou, and Christof Koch. The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature Reviews Neuroscience*, 13(6):407–420, 2012. doi: 10.1038/nrn3241.

Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat. Neurosci.*, 22(9):1512–1520, September 2019. doi: 10.1038/s41593-019-0460-x.

Georgia Christodoulou, Tim P. Vogels, and Everton J. Agnes. Regimes and mechanisms of transient amplification in abstract and biological neural networks. *PLOS Computational Biology*, 18(8): e1010365, 2022. doi: 10.1371/journal.pcbi.1010365.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/b618c3210e934362ac261db280128c22-Abstract.html.

Pablo S. G. Cisneros, Sophia Voss, and Herbert Werner. Efficient nonlinear model predictive control via quasi-LPV representation. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 3216–3221, 2016. doi: 10.1109/CDC.2016.7798752.

Brandon S. Coventry, Georgia L. Lawlor, Christina B. Bagnati, Claudia Krogmeier, and Edward L. Bartlett. Characterization and closed-loop control of infrared thalamocortical stimulation produces spatially constrained single-unit responses. *PNAS Nexus*, 3(2):pgae082, 2024. doi: 10.1093/pnasnexus/pgae082. URL https://doi.org/10.1093/pnasnexus/pgae082.

Carlos E. Davila, David X. Wang, Maxwell Ritzer, Rosalyn Moran, and Bradley C. Lega. A control-theoretical system for modulating hippocampal gamma oscillations using stimulation of the posterior cingulate cortex. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 30:2242–2253, 2022. doi: 10.1109/TNSRE.2022.3192170. URL https://doi.org/10.1109/TNSRE.2022.3192170.

Kathryn A. Davis, Seth P. Devries, Abba Krieger, Temenuzhka Mihaylova, Daniela Minecan, Brian Litt, Joost B. Wagenaar, and William C. Stacey. The effect of increased intracranial EEG sampling rates in clinical practice. *Clinical Neurophysiology*, 129(2):360–367, 2018. doi: 10.1016/j.clinph.2017.10.039.

Laura N. Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363, 2024. doi: 10.1038/s41593-024-01668-6. URL https://doi.org/10.1038/s41593-024-01668-6.

Lea Duncker and Maneesh Sahani. Dynamics on the manifold: Identifying computational dynamical activity from neural population recordings. *Current Opinion in Neurobiology*, 70:163–170, 2021. doi: 10.1016/j.conb.2021.10.014. URL https://doi.org/10.1016/j.conb.2021.10.014.

Salvador Dura-Bernal, Kan Li, Samuel A. Neymotin, Joseph T. Francis, Jose C. Principe, and William W. Lytton. Restoring behavior via inverse neurocontroller in a lesioned cortical spiking model driving a virtual arm. *Frontiers in Neuroscience*, 10:28, February 2016. doi: 10.3389/fnins.2016.00028. URL https://doi.org/10.3389/fnins.2016.00028.

Eray Erturk and Maryam M. Shanechi. Dynamical modeling of nonlinear latent factors in multiscale neural activity with real-time inference. In *Advances in Neural Information Processing Systems*, volume 38, pp. 31271–31310, 2025. doi: 10.52202/085713-0927. URL https://papers.nips.cc/paper_files/paper/2025/file/27e41a1f72811ce319f33e945cb45e6c-Paper-Conference.pdf.

Hao Fang, Stephen A. Berman, Yueming Wang, and Yuxiao Yang. Robust adaptive deep brain stimulation control of in-silico non-stationary Parkinsonian neural oscillatory dynamics. *Journal of Neural Engineering*, 21(3):036043, 2024. doi: 10.1088/1741-2552/ad5406. URL https://doi.org/10.1088/1741-2552/ad5406.

Christof Fehrman and C Daniel Meliza. Nonlinear model predictive control of a conductance-based neuron model via data-driven forecasting. *Journal of Neural Engineering*, 21(5):056014, 2024. ISSN 1741-2552. doi: 10.1088/1741-2552/ad731f. URL https://doi.org/10.1088/1741-2552/ad731f.

Christof Fehrman and C. Daniel Meliza. Model predictive control on the neural manifold. *Neural Computation*, 37(12):2125–2157, 2025. doi: 10.1162/neco.a.37. URL https://direct.mit.edu/neco/article/37/12/2125/133751/Model-Predictive-Control-on-the-Neural-Manifold.

Barbara Feulner and Claudia Clopath. Neural manifold under plasticity in a goal driven learning behaviour. *PLOS Computational Biology*, 17(2):e1008621, 2021. doi: 10.1371/journal.pcbi.1008621. URL https://doi.org/10.1371/journal.pcbi.1008621.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3601–3610, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7b7a53e239400a13bd6be6c91c4f6c4e-Abstract.html.

Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, June 2017. doi: 10.1016/j.neuron.2017.05.025.

Surya Ganguli, Dongsung Huh, and Haim Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975, 2008. doi: 10.1073/pnas.0804451105.

Qitong Gao, Stephen L. Schmidt, Afsana Chowdhury, Guangyu Feng, Jennifer J. Peters, Katherine Genty, Warren M. Grill, Dennis A. Turner, and Miroslav Pajic. Offline learning of closed-loop deep brain stimulation controllers for Parkinson disease treatment. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems*, pp. 44–55, 2023. doi: 10.1145/3576841.3585925. URL https://doi.org/10.1145/3576841.3585925.

Richard Gao, Ruud L. van den Brink, Thomas Pfeffer, and Bradley Voytek. Neuronal timescales are functionally dynamic and shaped by cortical microarchitecture. *eLife*, 9:e61277, 2020. doi: 10.7554/eLife.61277.

Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems*, volume 29, pp. 163–171, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/76dc611d6ebaafc66cc0879c71b5db5c-Abstract.html.

Charles M. Gaona, Mohit Sharma, Zachary V. Freudenburg, Jonathan D. Breshears, David T. Bundy, Jarod Roland, Dennis L. Barbour, Gerwin Schalk, and Eric C. Leuthardt. Nonuniform High-Gamma (60–500 Hz) Power Changes Dissociate Cognitive Task and Anatomy in Human Cortex. *Journal of Neuroscience*, 31(6):2091–2100, February 2011. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.4722-10.2011. URL https://www.jneurosci.org/content/31/6/2091.

Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, Feb 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04268-7. URL https://doi.org/10.1038/s41586-021-04268-7.

R Bhushan Gopaluni, Rohit S Patwardhan, and Sirish L Shah. The nature of data pre-filters in MPC relevant identification–open- and closed-loop issues. *Automatica*, 39(9):1617–1626, 2003. doi: 10.1016/S0005-1098(03)00146-8.

P. Gorzelic, Steven J. Schiff, and Alok Sinha. Model-based rational feedback controller design for closed-loop deep brain stimulation of Parkinson's disease. *Journal of Neural Engineering*, 10(2):026016, 2013. doi: 10.1088/1741-2560/10/2/026016. URL https://doi.org/10.1088/1741-2560/10/2/026016.

Roberto Guidotti, Alessio Basti, Giulia Pieramico, Antea D'Andrea, Saeed Makkinayeri, Mauro Pettorruso, Timo Roine, Ulf Ziemann, Risto J. Ilmoniemi, Gian Luca Romani, Vittorio Pizzella, and Laura Marzetti. When neuromodulation met control theory. *Journal of Neural Engineering*, 22(1):011001, 2025. doi: 10.1088/1741-2552/ad9958. URL https://doi.org/10.1088/1741-2552/ad9958.

Andrew Haddock, Anca Velisar, Jeffrey Herron, Helen Bronte-Stewart, and Howard Chizeck. Model predictive control of deep brain stimulation for Parkinsonian tremor. In *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 358–362. IEEE, 05 2017. doi: 10.1109/NER.2017.8008364. URL https://doi.org/10.1109/NER.2017.8008364.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, volume 97, pp. 2555–2565. PMLR, 2019.

Guillaume Hennequin, Tim P. Vogels, and Wulfram Gerstner. Non-normal amplification in random balanced neuronal networks. *Physical Review E*, 86(1):011909, 2012. doi: 10.1103/PhysRevE.86.011909.

Guillaume Hennequin, Tim P. Vogels, and Wulfram Gerstner. Optimal Control of Transient Dynamics in Balanced Networks Supports Generation of Complex Movements. *Neuron*, 82(6):1394–1406, June 2014. ISSN 0896-6273. doi: 10.1016/j.neuron.2014.04.045. URL https://www.cell.com/neuron/abstract/S0896-6273(14)00360-2.

Håkan Hjalmarsson. From experiment design to closed-loop control. *Automatica*, 41(3):393–438, 2005. doi: 10.1016/j.automatica.2004.11.021.

Thomas Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

Hari Teja Kalidindi, Kevin P. Cross, Timothy P. Lillicrap, Mohsen Omrani, Egidio Falotico, Philip N. Sabes, and Stephen H. Scott. Rotational dynamics in motor cortex are consistent with a feedback controller. *eLife*, 10:e67256, 2021. doi: 10.7554/eLife.67256. URL https://doi.org/10.7554/eLife.67256.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick Van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyTqHL5xg.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4743–4751,

2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/ddeebdeefdb7e7e7a697e1c3e3d8ef54-Abstract.html.

Thomas Koeglsperger, Maximilian Scherer, Andrea A. Kühn, Alexandre Eusebio, Alfonso Fasano, and Joachim K. Krauss. Adaptive deep brain stimulation for Parkinson's disease: Navigating the roadblocks to clinical implementation. *Movement Disorders*, 41(6):1370–1383, 2026. doi: 10.1002/mds.70284. URL https://doi.org/10.1002/mds.70284.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Deep Kalman filters. *arXiv preprint arXiv:1511.05121*, 2015. URL https://arxiv.org/abs/1511.05121.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770, 2020. URL https://proceedings.mlr.press/v120/lambert20a.html.

Chenguang Li, Gabriel Kreiman, and Sharad Ramanathan. Discovering neural policies to drive behaviour by integrating deep reinforcement learning agents with biological neural networks. *Nature Machine Intelligence*, 6(6):726–738, 2024. doi: 10.1038/s42256-024-00854-2. URL https://doi.org/10.1038/s42256-024-00854-2.

Zhichao Liang, Zixiang Luo, Keyin Liu, Jingwei Qiu, and Quanying Liu. Online learning Koopman operator for closed-loop electrical neurostimulation in epilepsy. *IEEE Journal of Biomedical and Health Informatics*, 27(1):492–503, 2023. doi: 10.1109/JBHI.2022.3210303. URL https://doi.org/10.1109/JBHI.2022.3210303.

Simon Little, Alex Pogosyan, Spencer Neal, Baltazar Zavala, Ludvic Zrinzo, Marwan Hariz, Thomas Foltynie, Patricia Limousin, Keyoumars Ashkan, James FitzGerald, Alexander L Green, Tipu Z Aziz, and Peter Brown. Adaptive deep brain stimulation in advanced Parkinson disease. *Ann. Neurol.*, 74(3):449–457, September 2013. doi: 10.1002/ana.23951. URL https://doi.org/10.1002/ana.23951.

Jianbo Liu, Hassan K Khalil, and Karim G Oweiss. Model-based analysis and control of a network of basal ganglia spiking neurons in the normal and Parkinsonian states. *Journal of Neural Engineering*, 8(4):045002, July 2011. ISSN 1741-2552. doi: 10.1088/1741-2560/8/4/045002. URL https://doi.org/10.1088/1741-2560/8/4/045002.

Andres M. Lozano, Nir Lipsman, Hagai Bergman, Peter Brown, Stephan Chabardes, Jin Woo Chang, Keith Matthews, Cameron C. McIntyre, Thomas E. Schlaepfer, Michael Schulder, Yasin Temel, Jens Volkmann, and Joachim K. Krauss. Deep brain stimulation: Current challenges and future directions. *Nature Reviews Neurology*, 15(3):148–160, 2019. doi: 10.1038/s41582-018-0128-2. URL https://doi.org/10.1038/s41582-018-0128-2.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. ISSN 0005-1098. doi: 10.1016/S0005-1098(99)00214-9. URL https://www.sciencedirect.com/science/article/pii/S0005109899002149.

Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016. doi: 10.1109/MCS.2016.2602087.

Jonathan A. Michaels, Benjamin Dann, and Hansjörg Scherberger. Neural population dynamics during reaching are better explained by a dynamical system than representational tuning. *PLOS Computational Biology*, 12(11):e1005175, 2016. doi: 10.1371/journal.pcbi.1005175. URL https://doi.org/10.1371/journal.pcbi.1005175.

Jonathan A. Michaels, Stefan Schaffelhofer, Andres Agudelo-Toro, and Hansjörg Scherberger. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proceedings of the National Academy of Sciences*, 117(50):32124–32135, 2020. doi: 10.1073/pnas.2005087117. URL https://www.pnas.org/doi/abs/10.1073/pnas.2005087117.

Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *eLife*, 6:e20899, 2017. doi: 10.7554/eLife.20899. URL https://doi.org/10.7554/eLife.20899.

Yuki Minai, Joana Soldado-Magraner, Matthew A. Smith, and Byron M. Yu. MiSO: Optimizing brain stimulation to create neural population activity states. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24126–24149, December 2024. doi: 10.52202/079017-0760. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2af641762dc02035c31a9314b2d090b6-Abstract-Conference.html.

Yuki Minai, Joana Soldado-Magraner, Byron M. Yu, and Matthew A. Smith. OMiSO: Adaptive optimization of state-dependent brain stimulation to shape neural population states. In *Advances in Neural Information Processing Systems*, volume 38, pp. 158197–158221, 2025. doi: 10.52202/085713-4771. URL https://papers.nips.cc/paper_files/paper/2025/hash/d1c1f61023dca672117b58f813a12d99-Abstract-Conference.html.

Brian A. Mitchell and Linda R. Petzold. Control of neural systems at multiple scales using model-free, deep reinforcement learning. *Scientific Reports*, 8(1):10721, 2018. doi: 10.1038/s41598-018-29134-x. URL https://doi.org/10.1038/s41598-018-29134-x.

Jeremy Morton, Freddie D. Witherden, Antony Jameson, and Mykel J. Kochenderfer. Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems*, volume 31, pp. 9258–9268, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2b0aa0d9e30ea3a55fc271ced8364536-Abstract.html.

Pehuén Moure, Jacob Granley, Fabrizio Grani, Leili Soo, Antonio Lozano, Rocío López-Peco, Adrián Villamarín-Ortiz, Cristina Soto-Sánchez, Shih-Chii Liu, Michael Beyeler, and Eduardo Fernández. Deep learning-based control of electrically evoked activity in human visual cortex. *Neuron*, August 2026. ISSN 0896-6273. doi: 10.1016/j.neuron.2026.07.006. URL https://www.cell.com/neuron/abstract/S0896-6273(26)00535-0.

Brendan K. Murphy and Kenneth D. Miller. Balanced amplification: A new mechanism of selective amplification of neural activity patterns. *Neuron*, 61(4):635–648, 2009. doi: 10.1016/j.neuron.2009.02.005.

John D. Murray, Alberto Bernacchia, David J. Freedman, Ranulfo Romo, Jonathan D. Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, and Xiao-Jing Wang. A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12):1661–1663, 2014. doi: 10.1038/nn.3862.

Kenneth R. Muske and James B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993. doi: 10.1002/aic.690390208. URL https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690390208.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7559–7566. IEEE, 2018. doi: 10.1109/ICRA.2018.8463189.

Amirhossein Nazerian, Malbor Asllani, Melvyn Tyloo, Wai Lim Ku, and Francesco Sorrentino. The frequency response of networks as open systems. *Nature Communications*, 17(1):2088, January 2026. ISSN 2041-1723. doi: 10.1038/s41467-026-68602-1. URL https://www.nature.com/articles/s41467-026-68602-1.

Carina R. Oehrn, Stephanie Cernera, Lauren H. Hammer, Maria Shcherbakova, Jiaang Yao, Amelia Hahn, Sarah Wang, Jill L. Ostrem, Simon Little, and Philip A. Starr. Chronic adaptive deep brain stimulation versus conventional stimulation in Parkinson's disease: a blinded randomized feasibility trial. *Nature Medicine*, 30(11):3345–3356, 2024. doi: 10.1038/s41591-024-03196-z. URL https://doi.org/10.1038/s41591-024-03196-z.

Lucine L. Oganesian and Maryam M. Shanechi. Brain–computer interfaces for neuropsychiatric disorders. *Nature Reviews Bioengineering*, 2(8):653–670, 2024. doi: 10.1038/s44222-024-00177-2. URL https://www.nature.com/articles/s44222-024-00177-2.

Daniel J. O'Shea, Lea Duncker, Werapong Goo, Xulu Sun, Saurabh Vyas, Eric M. Trautmann, Ilka Diester, Charu Ramakrishnan, Karl Deisseroth, Maneesh Sahani, and Krishna V. Shenoy. Direct neural perturbations reveal a dynamical mechanism for robust computation. *bioRxiv*, 2022. doi: 10.1101/2022.12.16.520768. URL https://doi.org/10.1101/2022.12.16.520768. Preprint.

Michelle Pan, Mariah L. Schrum, Vivek Myers, Erdem Biyik, and Anca Dragan. Coprocessor Actor Critic: A Model-Based Reinforcement Learning Approach For Adaptive Brain Stimulation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39292–39307. PMLR, July 2024. URL https://proceedings.mlr.press/v235/pan24g.html.

Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10): 805–815, 2018. doi: 10.1038/s41592-018-0109-9. URL https://doi.org/10.1038/s41592-018-0109-9.

Diogo Peixoto, Jessica R. Verhein, Roozbeh Kiani, Jonathan C. Kao, Paul Nuyujukian, Chandramouli Chandrasekaran, Julian Brown, Sania Fong, Stephen I. Ryu, Krishna V. Shenoy, and William T. Newsome. Decoding and perturbing decision states in real time. *Nature*, 591(7851): 604–609, 2021. doi: 10.1038/s41586-020-03181-9. URL https://doi.org/10.1038/s41586-020-03181-9.

Jaderson G Polli, Florian Kolbl, M G E da Luz, and P Lanusse. Delay suppression control of β-oscillations: a proposal for dual-target adaptive deep brain stimulation on STN-GPe network model. *Biol. Cybern.*, 119(4-6):21, July 2025. doi: 10.1007/s00422-025-01021-5. URL https://doi.org/10.1007/s00422-025-01021-5.

James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2 edition, 2017. ISBN 978-0-9759377-8-5. URL https://sites.engineering.ucsb.edu/~jbraw/mpc/.

Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):140–149, 2021. doi: 10.1038/s41593-020-00733-0. URL https://www.nature.com/articles/s41593-020-00733-0.

Omid G. Sani, Bijan Pesaran, and Maryam M. Shanechi. Dissociative and prioritized modeling of behaviorally relevant neural dynamics using recurrent neural networks. *Nature Neuroscience*, 27(10):2033–2045, 2024. doi: 10.1038/s41593-024-01731-2. URL https://www.nature.com/articles/s41593-024-01731-2.

Sabato Santaniello, Giovanni Fiengo, Luigi Glielmo, and Warren M. Grill. Closed-Loop Control of Deep Brain Stimulation: A Simulation Study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(1):15–24, February 2011. ISSN 1558-0210. doi: 10.1109/TNSRE.2010.2081377. URL https://ieeexplore.ieee.org/abstract/document/5590300.

John D. Simeral, Thomas Hosman, Jad Saab, Sharlene N. Flesher, Marco Vilela, Brian Franco, Jessica N. Kelemen, David M. Brandman, John G. Ciancibello, Paymon G. Rezaii, Emad N. Eskandar, David M. Rosler, Krishna V. Shenoy, Jaimie M. Henderson, Arto V. Nurmikko, and Leigh R. Hochberg. Home use of a percutaneous wireless intracortical brain-computer interface by individuals with tetraplegia. *IEEE Transactions on Biomedical Engineering*, 68(7):2313–2325, 2021. doi: 10.1109/TBME.2021.3069119. URL https://doi.org/10.1109/TBME.2021.3069119.

Sebastian Steffen and Mark Cannon. Deep learning model predictive control for deep brain stimulation in Parkinson's disease. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pp. 5744–5749. IEEE, 2025. doi: 10.1109/CDC57313.2025.11312469. URL https://doi.org/10.1109/CDC57313.2025.11312469.

Sebastian Steffen, Mark Cannon, Huiling Tan, and Jean Debarros. Model predictive control for closed-loop deep brain stimulation. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 4034–4039, 2024. doi: 10.1109/CDC56724.2024.10885888. URL https://doi.org/10.1109/CDC56724.2024.10885888.

Robert F. Stengel. *Optimal Control and Estimation*. Dover Books on Mathematics. Dover Publications, 2012. ISBN 9780486134819. URL https://store.doverpublications.com/products/9780486134819.

Fei Su, Jiang Wang, Shuangxia Niu, Huiyan Li, Bin Deng, Chen Liu, and Xile Wei. Nonlinear predictive control for adaptive adjustments of deep brain stimulation parameters in basal ganglia–thalamic network. *Neural Networks*, 98:283–295, 2018. doi: 10.1016/j.neunet.2017.12.001. URL https://doi.org/10.1016/j.neunet.2017.12.001.

David Sussillo, Mark M. Churchland, Matthew T. Kaufman, and Krishna V. Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, 18(7): 1025–1033, 2015. doi: 10.1038/nn.4042. URL https://doi.org/10.1038/nn.4042.

Sina Tafazoli, Camden J MacDowell, Zongda Che, Katherine C Letai, Cynthia R Steinhardt, and Timothy J Buschman. Learning to control the brain through adaptive closed-loop patterned stimulation. *Journal of Neural Engineering*, 17(5):056007, October 2020. ISSN 1741-2552. doi: 10.1088/1741-2552/abb860. URL https://dx.doi.org/10.1088/1741-2552/abb860.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. doi: 10.1126/science.290.5500.2319.

Parsa Vahidi, Omid G Sani, and Maryam M Shanechi. Modeling and dissociation of intrinsic and input-driven neural population dynamics underlying behavior. *Proceedings of the National Academy of Sciences*, 121(7):e2212887121, 2024. doi: 10.1073/pnas.2212887121.

Peter Van Overschee and Bart De Moor. *Subspace Identification for Linear Systems: Theory—Implementation—Applications*. Kluwer Academic Publishers, Boston, MA, 1996. doi: 10.1007/978-1-4613-0465-4.

Anca Velisar, Judy Syrkin-Nikolau, Zack Blumenfeld, Matthew H. Trager, Mohammad F. Afzal, Vinay Prabhakar, and Helen Bronte-Stewart. Dual threshold neural closed loop deep brain stimulation in Parkinson disease patients. *Brain Stimulation*, 12(4):868–876, 2019. doi: 10.1016/j.brs.2019.02.020. URL https://doi.org/10.1016/j.brs.2019.02.020.

Amit Vinograd, Aditya Nair, Joseph H. Kim, Scott W. Linderman, and David J. Anderson. Causal evidence of a line attractor encoding an affective state. *Nature*, 634(8035):910–918, 2024. doi: 10.1038/s41586-024-07915-x. URL https://doi.org/10.1038/s41586-024-07915-x.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2746–2754, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/a1afc58c6ca9540d057299ec3016d726-Abstract.html.

Marco Weiergräber, Anna Papazoglou, Karl Broich, and Ralf Müller. Sampling rate, signal bandwidth and related pitfalls in EEG analysis. *Journal of Neuroscience Methods*, 268:53–55, 2016. doi: 10.1016/j.jneumeth.2016.05.010.

Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, 2019. doi: 10.1038/s41593-018-0310-2. URL https://doi.org/10.1038/s41593-018-0310-2.

Yuxiao Yang, Allison T Connolly, and Maryam M Shanechi. A control-theoretic system identification framework and a real-time closed-loop clinical simulation testbed for electrical brain stimulation. *J. Neural Eng.*, 15(6):066007, December 2018. doi: 10.1088/1741-2552/aad1a8. URL https://doi.org/10.1088/1741-2552/aad1a8.

Yuxiao Yang, Shaoyu Qiao, Omid G. Sani, J. Isaac Sedillo, Breonna Ferrentino, Bijan Pesaran, and Maryam M. Shanechi. Modelling and prediction of the dynamic responses of large-scale brain networks during direct electrical stimulation. *Nature Biomedical Engineering*, 5 (4):324–345, 2021. doi: 10.1038/s41551-020-00666-w. URL https://doi.org/10.1038/s41551-020-00666-w.

Aarne Ylinen, Anatol Bragin, Zoltán Nádasdy, Gábor Jandó, Imre Szabó, Attila Sik, and György Buzsáki. Sharp wave–associated high-frequency oscillation (200 Hz) in the intact hippocampus: Network and intracellular mechanisms. *Journal of Neuroscience*, 15(1):30–46, 1995. doi: 10.1523/JNEUROSCI.15-01-00030.1995.

A ADDITIONAL RELATED WORK

Several further lines of work aim to influence neural activity with stimulation but do not control an evolving population state from an identified dynamical model.

One line of work designs feedback controllers directly on hand-specified biophysical models of disease circuits *in silico*, without learning a model from data. These circuits typically include basal ganglia-thalamocortical models of Parkinson's disease, regulating variables assumed directly accessible such as the firing rates, synaptic outputs, or beta-band power (Liu et al., 2011; Gorzelic et al., 2013; Polli et al., 2025; Fang et al., 2024). Because the model is specified rather than learned, these studies do not address identification from neural data.

A second line fits nonlinear models from data and optimizes stimulation against them numerically *in silico*, *e.g.*, for a single conductance-based neuron or for biomarkers such as beta oscillations (Su et al., 2018; Fehrman & Meliza, 2024; Steffen & Cannon, 2025). These studies forecast the signals without representing them with a compact latent population state or embedding (requirement 1), and without a filtering capability for noisy or missing observations (requirement 2). Furthermore, control must be computed by numerical optimization of a fully nonlinear model. We instead consider the distinct problem of learning a latent state from noisy, partially observed population activity that can be used for model-predictive control.

A third line of work picks the stimulation that will elicit a target neural response, often through a static inverse map. These works typically fix the stimulus within a trial or session and optimize which stimulus to deliver on a trial-by-trial or session-by-session basis. Stimulation patterns have been refined across trials by comparing the evoked response to a target (Tafazoli et al., 2020), or selected by inverting a model that maps stimulation parameters to the evoked response, in some cases conditioned on the pre-stimulation state (Minai et al., 2024; 2025; Moure et al., 2026), including in silico to recover pre-lesion activity in a lesioned spiking network (Dura-Bernal et al., 2016). A related approach identifies an input-response map once per session and inverts it to design the stimulation sequence (Barzon et al., 2026), while another fits a forward model predicting the response to stimulation from a short window of preceding activity and uses it in silico to decide whether to stimulate on each trial (Bryan et al., 2025). In all of these, the stimulus is chosen per trial or session from a response mapping, rather than computed continuously from a dynamic model of how the neural state is evolving.

B MANIFOLDS

B.1 MANIFOLDS A SWEEP

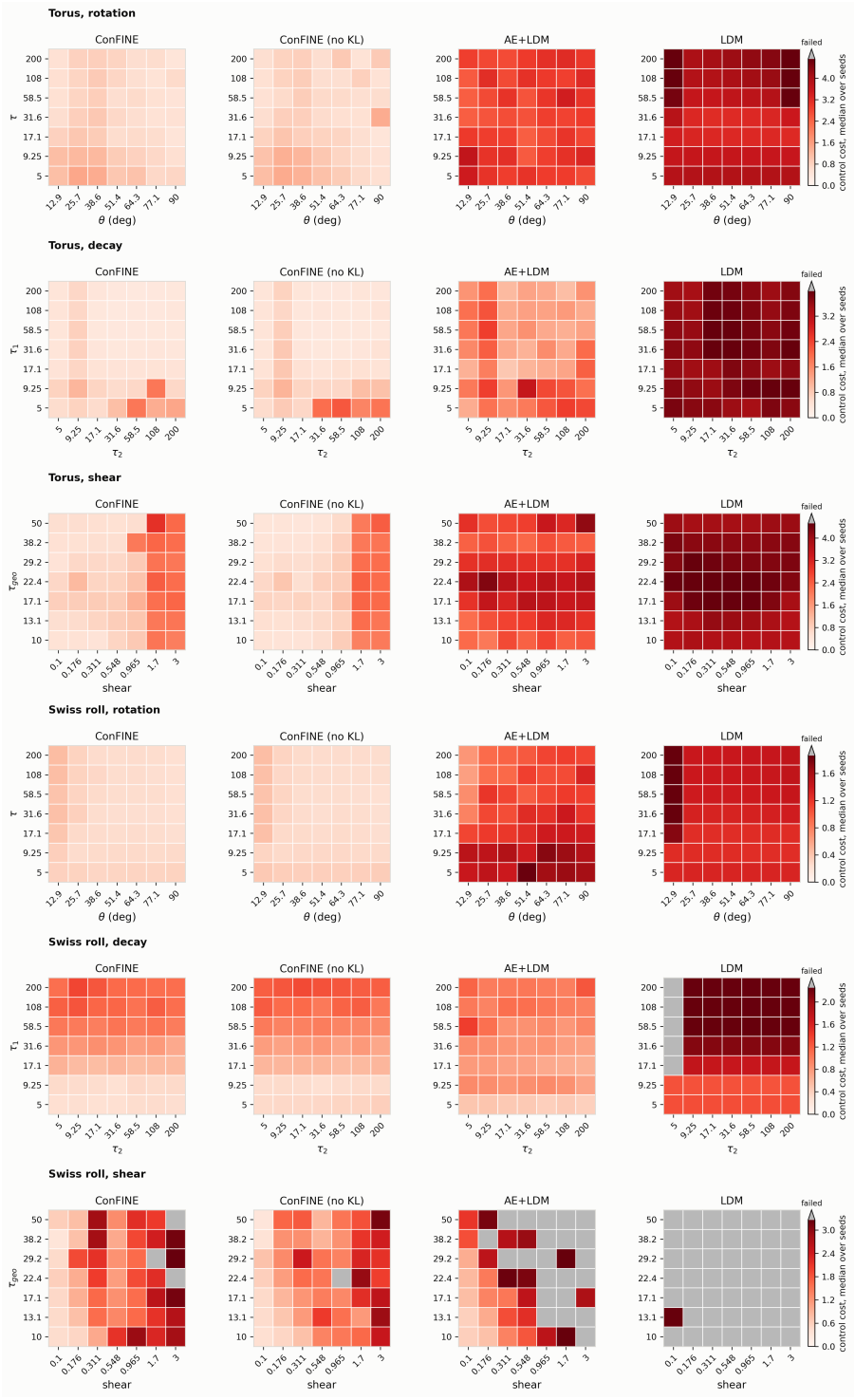

Figure A1: Control cost of every plant of the six two-dimensional families (rows) for each model (columns); the ring is in Fig. A2. Each cell is one plant, colored by its median cost over training seeds. Gray ("failed"): worse than chance. τ_{geo} is the geometric-mean time constant $\bar{\tau}$.

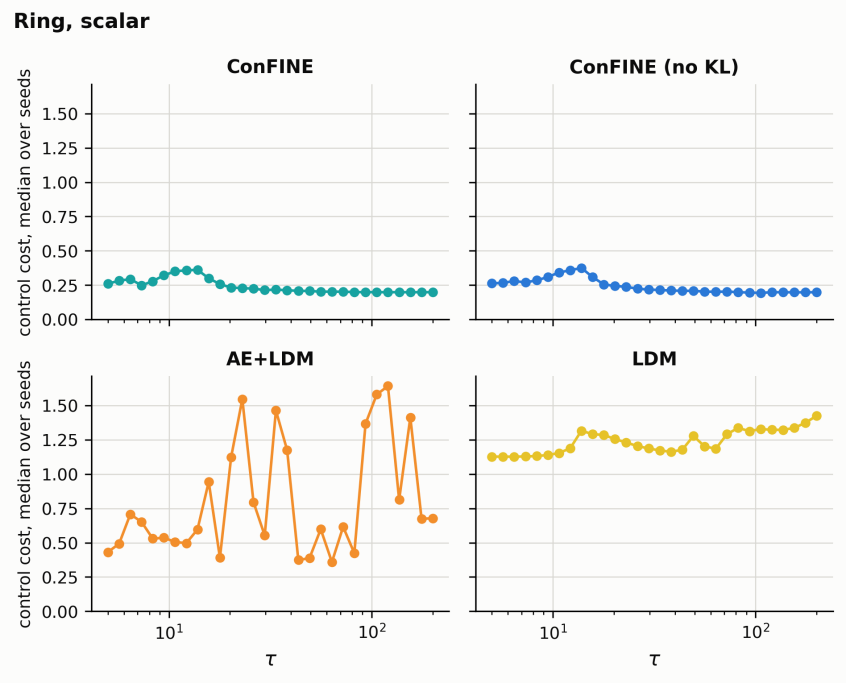

Figure A2: Control cost of each model on the 30 ring plants against the time constant τ, median over training seeds. No ring plant is worse than chance.

B.2 MANIFOLDS VIOLIN PLOTS

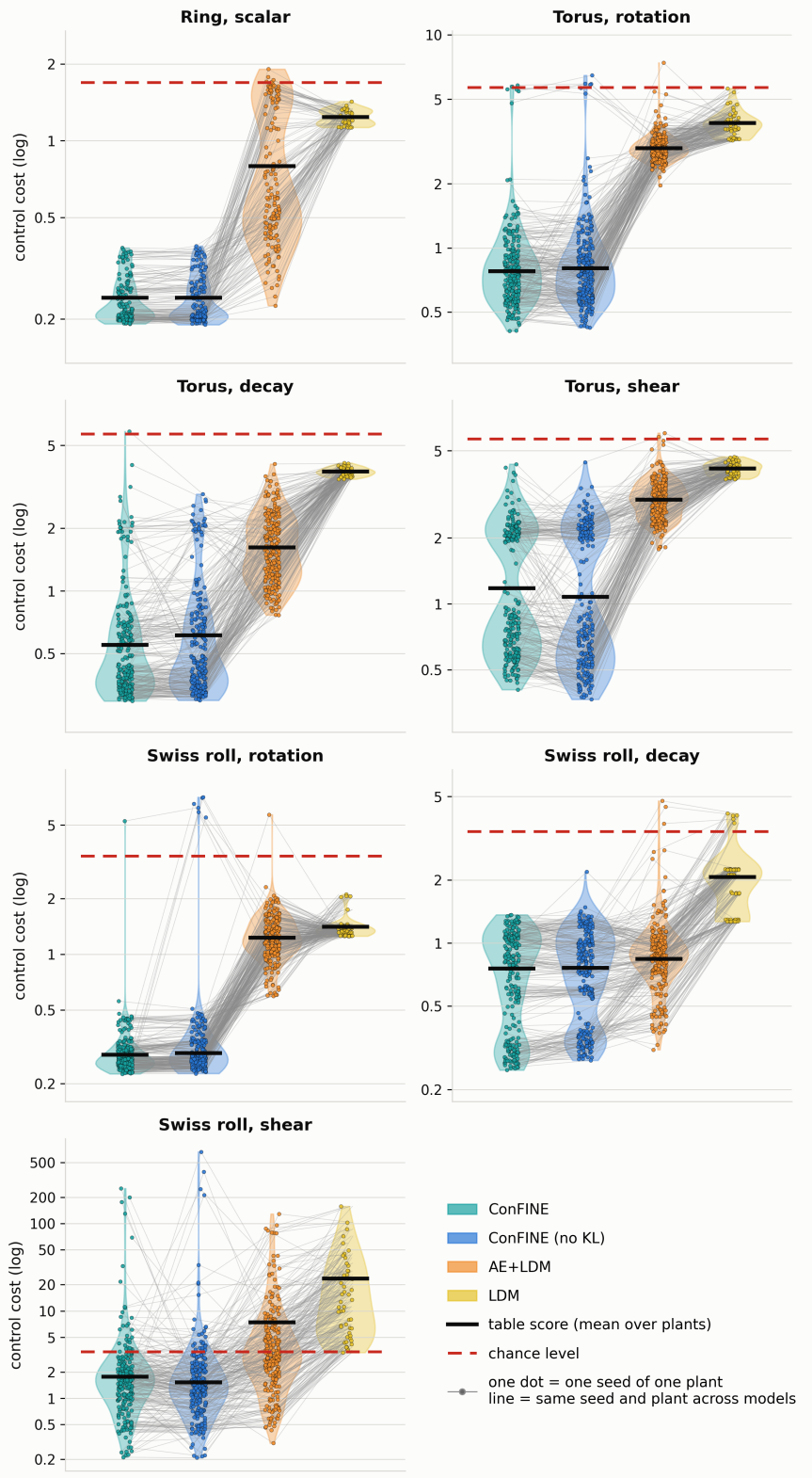

Figure A3: Control cost over all plants and training seeds of each family (log scale). Black bars are the scores of Tab. 1.

B.3 MANIFOLDS TRAJECTORIES

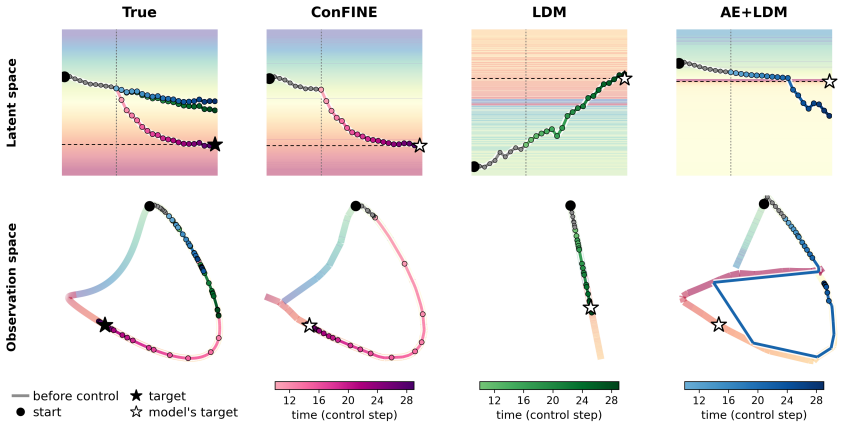

Figure A4: As Fig. 3, on a ring plant.

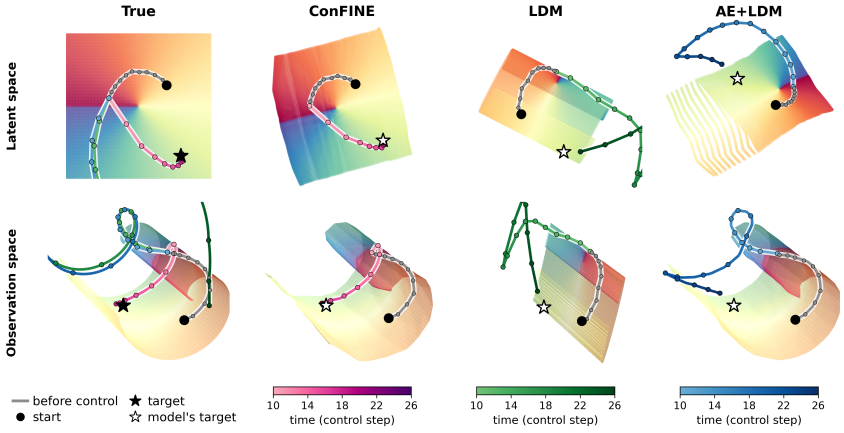

Figure A5: As Fig. 3, on a Swiss-roll rotation plant.

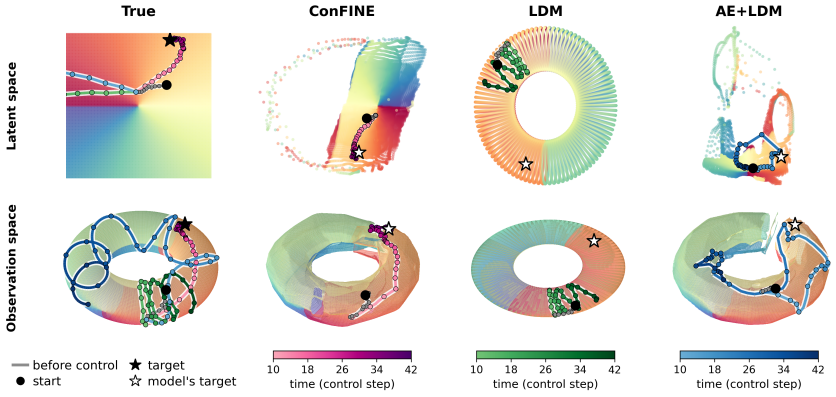

Figure A6: As Fig. 3, on a torus shear plant.

C RNN

C.1 Example Trajectories

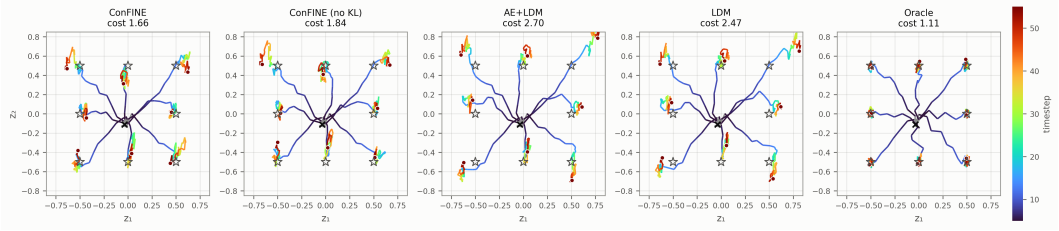

Figure A7: One closed-loop episode on the reach-trained RNN (plant B, $R = 10^{-3}$), one panel per model. All panels share the same episode: the same start (cross), plant-noise seed, and eight targets (stars). Color is the time step once control switches on at step 5; the preceding steps are gray, and the dot marks the final position. The cost above each panel is this episode's cost averaged over the eight targets. Oracle: MPC with the true plant and true state.

C.2 Bar plots

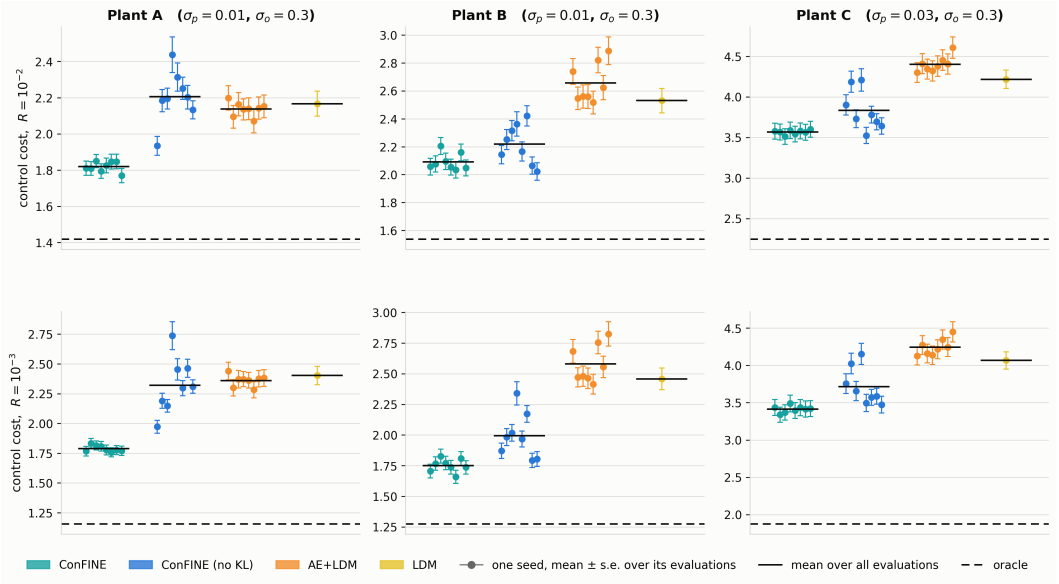

Figure A8: Per-seed control cost on the reach-trained RNN plants A, B, and C (columns) at $R = 10^{-2}$ and 10^{-3} (rows); each seed is evaluated on 25 episodes $\times$ 8 targets, and the mean over all evaluations is the number reported in Tab. 2. LDM has a single fit (App. E.2).

C.3 Dropping Samples Performance

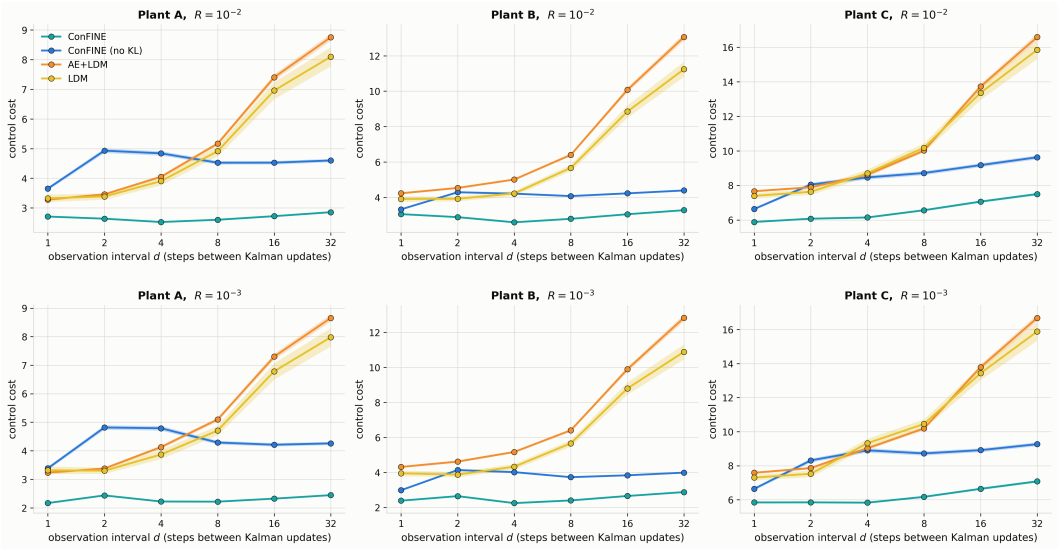

Figure A9: As Fig. 4, on all three RNN plants (columns) at $R = 10^{-2}$ (top) and 10^{-3} (bottom). Episodes are 105 steps rather than the 55 of Tab. 2, so costs are not comparable to the table.

D Stroke Recovery

D.1 Per seed performance

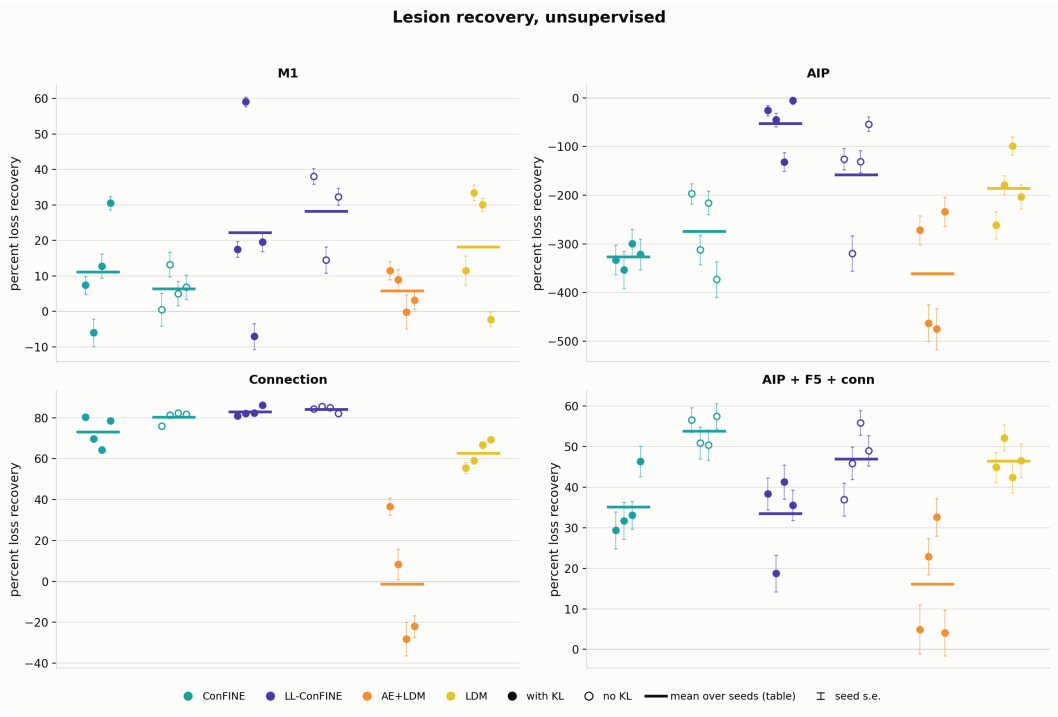

Figure A10: Per-seed percent recovery, unsupervised protocol: one marker per training seed, with its bootstrap s.e. over the 100 held-out trials; horizontal lines are the means over seeds reported in Tab. 3.

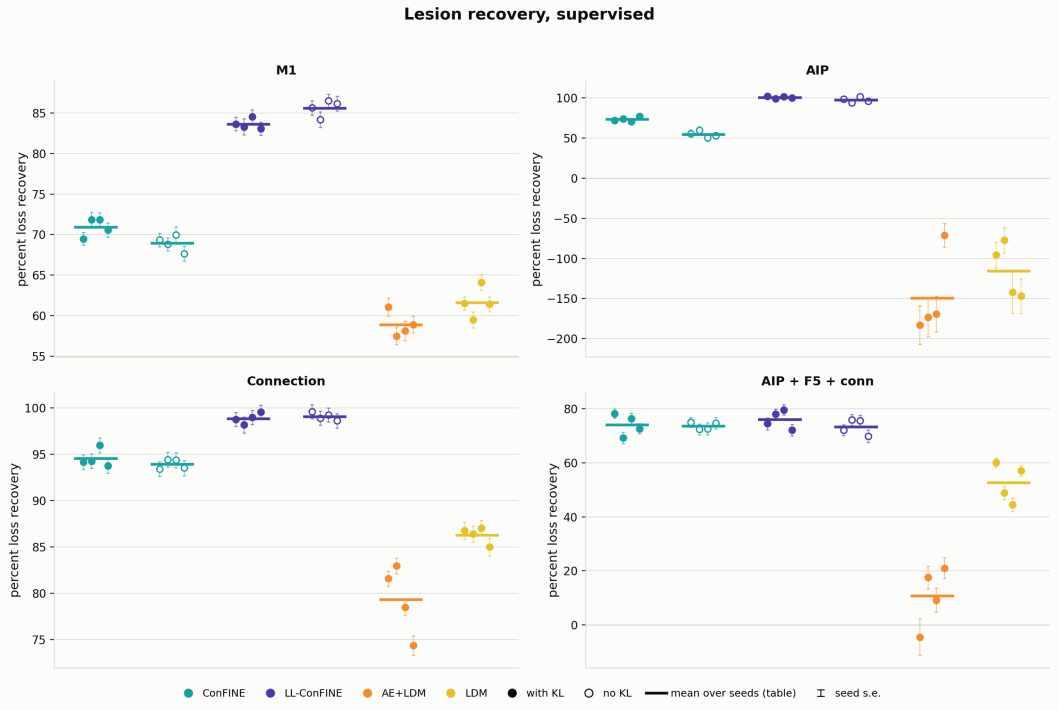

Figure A11: As Fig. A10, supervised protocol. On every lesion, every seed of every ConFINE variant recovers more than every seed of both baselines.

D.2 Pairwise Statistical Significance

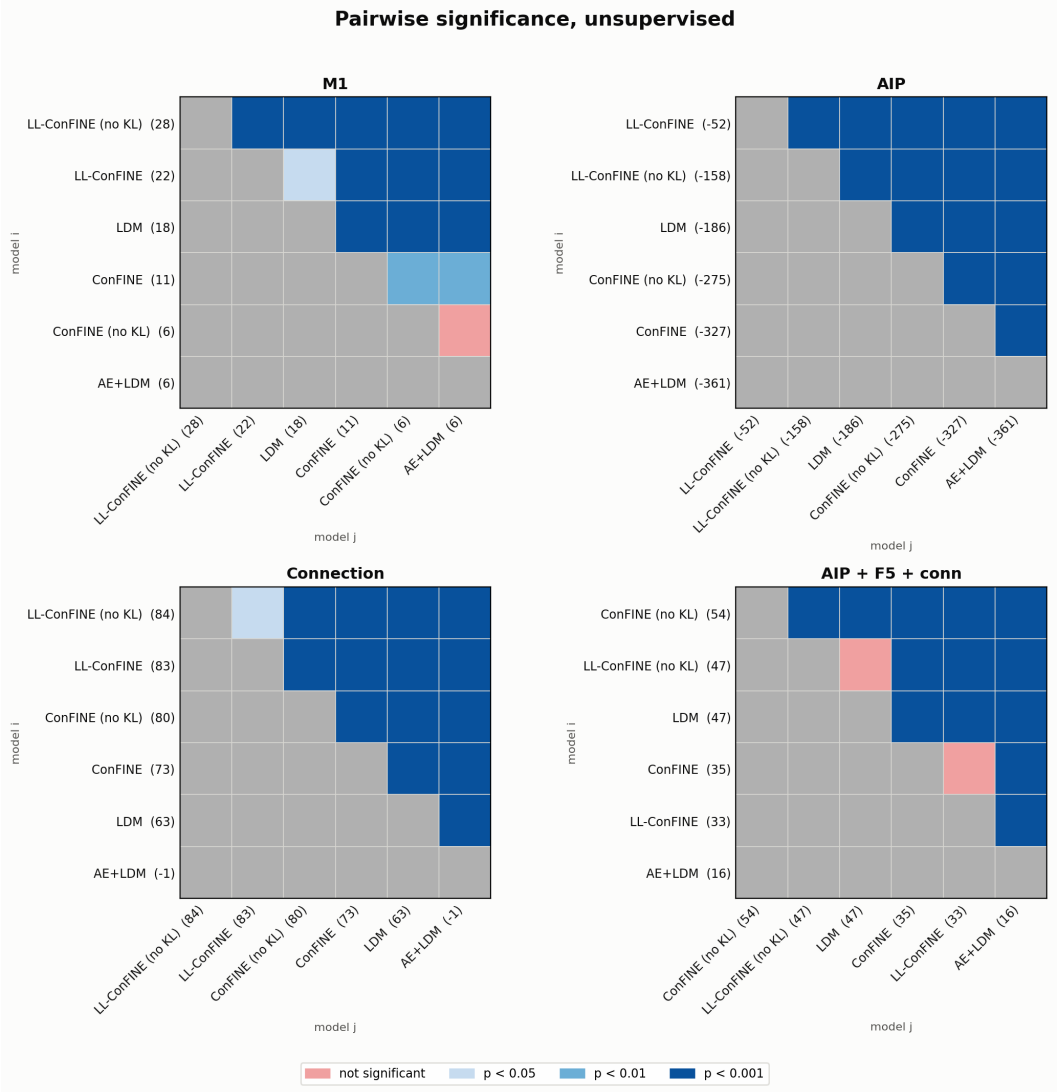

Figure A12: Pairwise comparison of the models of Tab. 3, unsupervised protocol, sorted by mean recovery (in parentheses). The cell in row i, column j tests whether model i has a better recovery than model j: one-sided Wilcoxon signed-rank test on per-trial losses, paired by trial and training seed.

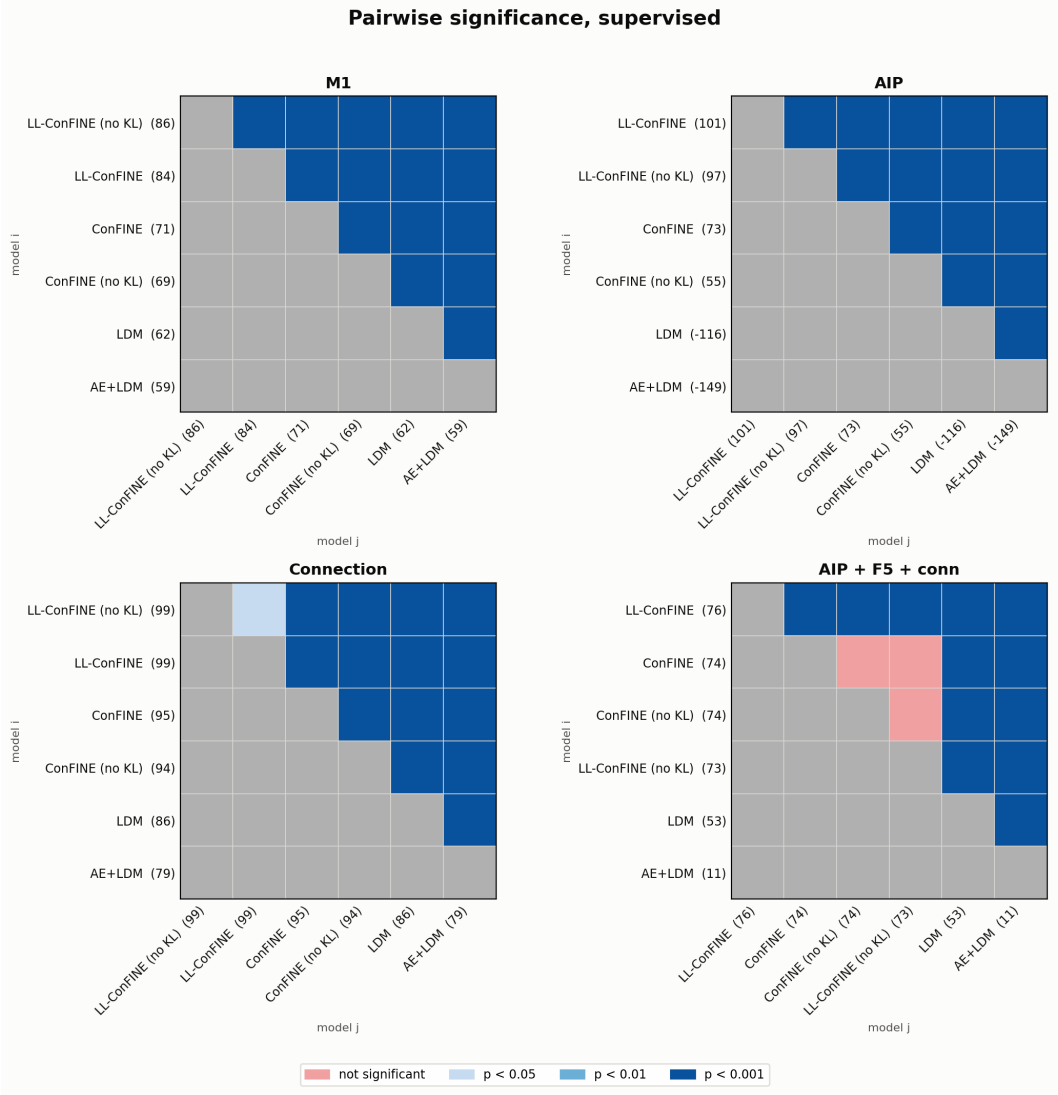

Figure A13: As Fig. A12, supervised protocol. Every ConFINE variant has a significantly lower loss than both baselines on every lesion ($p < 0.001$).

D.3 Observation/Behavior Prediction vs Control

For every trained model of Tab. 3, we compute the k-step prediction error of the neural observations y and of the behavior z on 256 new episodes of random stimulation, generated like the training data, as the mean squared error divided by the data variance (NMSE), averaged over $k = 1, \ldots, 8$. For unsupervised models, z is predicted through the post-hoc readout (9). Recovery tracks behavior prediction error (R from -0.90 to -0.67) but not observation prediction error (R from -0.40 to $+0.35$): the unsupervised models predict the observations best on every lesion, yet almost always recover less than their supervised counterparts.

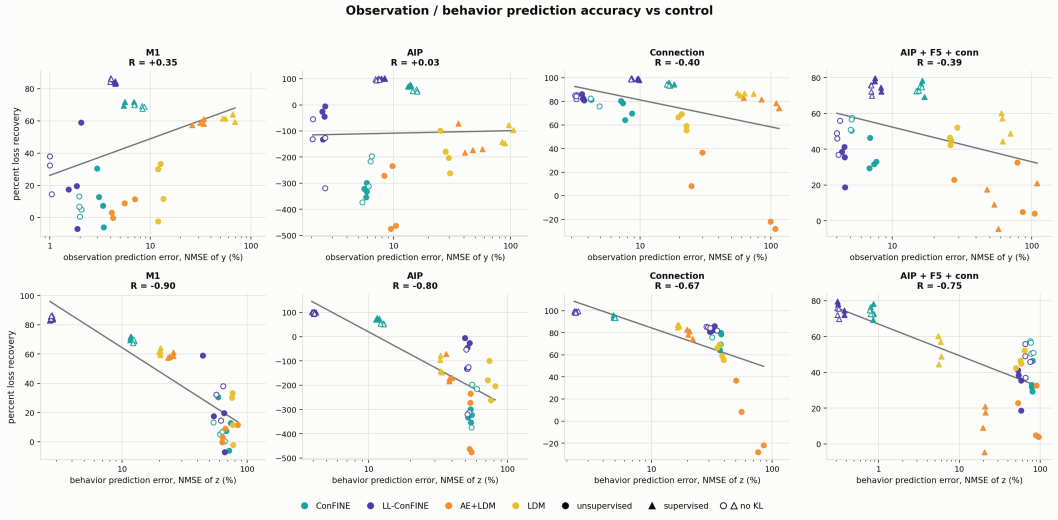

Figure A14: Prediction error versus percent recovery, one marker per trained model. Lines are least-squares fits against $\log_{10}$ NMSE; R is the Pearson correlation between recovery and $\log_{10}$ NMSE, pooled over both protocols.

E Model parameterization and training details

Notation. We denote the estimate of the state x_s given $\tilde{a}_{1:t}$ as $\mathcal{N}(\hat{x}_{s|t}, P_{s|t})$, and for any state-dependent term $M \in \{A, B, C, \Sigma_x, \Sigma_a, W_z, b_z\}$ we write $M_{s|t} := M(\hat{x}_{s|t})$ for the term evaluated at that estimate.

Generative model. Conditioned on the past, the model is a linear Gaussian state-space model whose matrices are set by the current belief about the state, i.e., a quasi-linear parameter-varying system:

$$x_{t+1} = A_{t|t}\, x_t + B_{t|t}\, u_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, \Sigma_{x,t|t}) \tag{15}$$

$$a_t = C_{t|t-1}\, x_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \Sigma_{a,t|t-1}) \tag{16}$$

$$y_t = f_{\mathrm{dec}}(a_t) \tag{17}$$

$$z_t = W_{z,t|t-1}\, x_t + b_{z,t|t-1} \tag{18}$$

with diagonal noise covariances

$$\Sigma_{x,s|t} = \mathrm{diag}\big(\sigma^2_{x,1,s|t}, \ldots, \sigma^2_{x,n_x,s|t}\big), \qquad \Sigma_{a,s|t} = \mathrm{diag}\big(\sigma^2_{a,1,s|t}, \ldots, \sigma^2_{a,n_a,s|t}\big).$$

The behavior readout (4) is present only when behavioral data are available. The *globally linear* model is the special case in which all of these terms are constant; all equations below hold unchanged with the subscripts on the matrices dropped.

Encoder and decoder. The encoder is a multilayer perceptron $f_{\mathrm{enc}} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_a}$ with the hidden widths and activation listed in Tab. 4. The decoder $f_{\mathrm{dec}} : \mathbb{R}^{n_a} \to \mathbb{R}^{n_y}$ mirrors it (same widths in reverse order, same activation). Observations are standardized per channel with training-set statistics before the encoder and de-standardized after the decoder; inputs u are not scaled. LDM uses the identity encoder and decoder, so $a = y$ (standardized) and $n_a = n_y$.

Dynamics and emission. In the globally linear model A, B and C are free matrices, initialized as $A = I$ (except on the manifolds; App. E.2) and B, C with standard normal entries. The noise covariances are diagonal,

$$\sigma^2_{\cdot,i} = \sigma^2_{\min} + \mathrm{softplus}(v_i), \qquad \sigma^2_{\min} = 10^{-2}, \tag{19}$$

with v a learned vector initialized with standard normal entries.

Locally linear terms. In the locally linear model the state-dependent terms are linear heads on a shared two-layer ReLU network $f(\cdot)$ with 128 hidden units, applied to the belief mean:

$$A(x) = I + \gamma \,\mathrm{vec}^{-1}\big(W_A f(x)\big), \tag{20}$$

$$B(x) = \mathrm{vec}^{-1}\big(W_B f(x)\big), \tag{21}$$

$$C(x) = \mathrm{vec}^{-1}\big(W_C f(x)\big), \tag{22}$$

$$\sigma^2_{x,i}(x) = \sigma^2_{\min} + \mathrm{softplus}\big(W_{\sigma_x} f(x)\big)_i, \tag{23}$$

$$\sigma^2_{a,i}(x) = \sigma^2_{\min} + \mathrm{softplus}\big(W_{\sigma_a} f(x)\big)_i, \tag{24}$$

where γ is a learnable scalar initialized at 10^{-2} so that A starts near the identity. The hidden layers of f use orthogonal initialization with zero biases. Everything else is identical to the globally linear model with the same encoder.

Behavior readout. In all our experiments, the readout is a single linear layer $z = W_z x + b_z$ with constant W_z, b_z, including in the locally linear model; in general, W_z and b_z can also be state-dependent heads on $f(\cdot)$, like the other terms. In the supervised protocol, the readout is trained jointly with the model on the filtered state $\hat{x}_{t|t}$ and on the predicted states $\hat{x}_{t+k|t}$, $k \in \mathcal{K}_z$; in the unsupervised protocol, it is fitted by least squares on the same states of the trained model.

E.1 Inference and prediction

Encoding.

$$\tilde{a}_t = f_{\mathrm{enc}}(y_t) \tag{25}$$

Prediction (one step). Initialized with $\hat{x}_{1|0} = 0$ and $P_{1|0} = I$:

$$p(x_t|\tilde{a}_{1:t-1}) = \mathcal{N}(x_t; \hat{x}_{t|t-1}, P_{t|t-1}) \tag{26}$$

$$\hat{x}_{t|t-1} = A_{t-1|t-1}\,\hat{x}_{t-1|t-1} + B_{t-1|t-1}\,u_{t-1} \tag{27}$$

$$P_{t|t-1} = A_{t-1|t-1} P_{t-1|t-1} A^\top_{t-1|t-1} + \Sigma_{x,t-1|t-1} \tag{28}$$

Update.

$$p(x_t|\tilde{a}_{1:t}) = \mathcal{N}(x_t; \hat{x}_{t|t}, P_{t|t}) \tag{29}$$

$$S_{t|t-1} = C_{t|t-1} P_{t|t-1} C^\top_{t|t-1} + \Sigma_{a,t|t-1} \tag{30}$$

$$K_t = P_{t|t-1} C^\top_{t|t-1} S^{-1}_{t|t-1} \tag{31}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(\tilde{a}_t - C_{t|t-1}\hat{x}_{t|t-1}) \tag{32}$$

$$P_{t|t} = (I - K_t C_{t|t-1}) P_{t|t-1} \tag{33}$$

Filtered estimates.

$$p(a_t|\tilde{a}_{1:t}) = \int p(a_t|x_t)\, p(x_t|\tilde{a}_{1:t})\, dx_t = \mathcal{N}(a_t; \hat{a}_{t|t}, S_{t|t}) \tag{34}$$

$$\hat{a}_{t|t} = C_{t|t}\,\hat{x}_{t|t} \tag{35}$$

$$S_{t|t} = C_{t|t} P_{t|t} C^\top_{t|t} + \Sigma_{a,t|t} \tag{36}$$

$$\hat{y}_{t|t} = f_{\mathrm{dec}}(\hat{a}_{t|t}) \tag{37}$$

$$\hat{z}_{t|t} = W_{z,t|t}\,\hat{x}_{t|t} + b_{z,t|t} \tag{38}$$

Prediction (k steps). The rollout starts from the filtered belief $\mathcal{N}(\hat{x}_{t|t}, P_{t|t})$ and, for $j = 1, \ldots, k$, evaluates every state-dependent term at the current predicted mean:

$$p(x_{t+j}|\tilde{a}_{1:t}) = \mathcal{N}(x_{t+j}; \hat{x}_{t+j|t}, P_{t+j|t}) \tag{39}$$

$$\hat{x}_{t+j|t} = A_{t+j-1|t}\,\hat{x}_{t+j-1|t} + B_{t+j-1|t}\,u_{t+j-1} \tag{40}$$

$$P_{t+j|t} = A_{t+j-1|t} P_{t+j-1|t} A^\top_{t+j-1|t} + \Sigma_{x,t+j-1|t} \tag{41}$$

$$p(a_{t+k}|\tilde{a}_{1:t}) = \int p(a_{t+k}|x_{t+k})\, p(x_{t+k}|\tilde{a}_{1:t})\, dx_{t+k} = \mathcal{N}(a_{t+k}; \hat{a}_{t+k|t}, S_{t+k|t}) \tag{42}$$

$$\hat{a}_{t+k|t} = C_{t+k|t}\, \hat{x}_{t+k|t} \tag{43}$$

$$S_{t+k|t} = C_{t+k|t} P_{t+k|t} C_{t+k|t}^{\top} + \Sigma_{a,t+k|t} \tag{44}$$

$$\hat{y}_{t+k|t} = f_{\mathrm{dec}}(\hat{a}_{t+k|t}) \tag{45}$$

$$\hat{z}_{t+k|t} = W_{z,t+k|t}\, \hat{x}_{t+k|t} + b_{z,t+k|t} \tag{46}$$

E.2 Training

Data. The observations y_t are z-scored before training the model and during inference; the inputs u_t are not.

Loss. The prediction loss uses the horizons $k = 0$ (filtering) to $k = 8$; the per-horizon weights, the KL weight and free nats α, and the behavior-readout weight are given in Tab. 4.

Initialization. On the manifolds, A is initialized $n_{\mathrm{seed}} = 3$ times; each candidate is trained for $E_{\mathrm{burn}} = 3$ epochs, and training continues from the candidate with the lowest prediction error. Each candidate is constructed as follows:

1. For candidate i, choose a fraction β_i of oscillatory versus real modes, with $\beta_1, \ldots, \beta_{n_{\mathrm{seed}}}$ spaced uniformly over $[0, 1]$.
2. For each real mode, draw the radius uniformly from $[0.8, 0.999]$; for each oscillatory mode, draw the radius from the same range and the angle from $[0.1, \pi/2]$ to form a 2×2 rotation block.
3. A is block-diagonal, interleaving the real and rotation blocks.

On the reach RNN and the grasping circuit, A is initialized to the identity.

Optimization. All gradient-trained models use Adam with cosine annealing of the learning rate to zero over the run, stepped once per update, and gradient-norm clipping at 1. Training uses random chunks: every batch element is a window of fixed length starting at a random offset of a training sequence.

Baselines. LDM uses the identity encoder and decoder. AE+LDM first trains the encoder and decoder for 25 epochs on reconstruction alone, then freezes them and fits a linear state-space model on the frozen latent. On the manifolds and the reach RNN, the linear state-space model of both baselines is fit by subspace identification (Van Overschee & De Moor, 1996), using IPSID (Vahidi et al., 2024) with no behavioral states and block-Hankel horizon 8; subspace identification is deterministic, so LDM has a single fit per plant. On the grasping circuit, both baselines are trained by gradient descent.

F Control

We use KFUpdate and KFPredict to denote the Kalman-filter recursions in App. E.1. Before control onset (Tab. 5), the filter runs with $u_t = 0$.

F.1 Observation targets

Given the target pair (x^*, u^*) from (8), and initializing the predicted trajectory from the current filtered estimate, the MPC solves the box-constrained quadratic program

$$\begin{aligned} \min_{u_{t:t+H-1}} \quad & \sum_{k=t}^{t+H-1} \left[(\hat{x}_{k+1|t} - x^*)^{\top} Q (\hat{x}_{k+1|t} - x^*) + (u_k - u^*)^{\top} R (u_k - u^*) \right] \\ \text{s.t.} \quad & \hat{x}_{k+1|t} = A \hat{x}_{k|t} + B u_k, \\ & u_{\min} \le u_k \le u_{\max}, \qquad k = t, \ldots, t+H-1. \end{aligned} \tag{47}$$

Algorithm 1 Control to an observation target y^*

Require: learned model $(A, B, C, \Sigma_x, \Sigma_a, f_{\text{enc}})$; target y^*; horizon H; weights Q, R; input box $[u_{\min}, u_{\max}]$; equilibrium weight λ_{eq}

$\tilde{a}^* \leftarrow f_{\text{enc}}(y^*)$
$(x^*, u^*) \leftarrow$ solution of (8)
$\hat{x}_{1|0} \leftarrow 0,\ P_{1|0} \leftarrow I$
for $t = 1, 2, \ldots$ **do**
  observe y_t
  $(\hat{x}_{t|t}, P_{t|t}) \leftarrow \text{KFUpdate}(\hat{x}_{t|t-1}, P_{t|t-1}, f_{\text{enc}}(y_t))$
  $u_{t:t+H-1} \leftarrow$ solution of (47)
  apply u_t
  $(\hat{x}_{t+1|t}, P_{t+1|t}) \leftarrow \text{KFPredict}(\hat{x}_{t|t}, P_{t|t}, u_t)$
end for

Algorithm 2 Control to a behavioral reference

Require: learned model $(A, B, C, \Sigma_x, \Sigma_a, f_{\text{enc}})$; behavioral readout (W_z, b_z); reference z^* or $\{z_t^*\}$; horizon H; weights Q, R; input box $[u_{\min}, u_{\max}]$

$\hat{x}_{1|0} \leftarrow 0,\ P_{1|0} \leftarrow I$
for $t = 1, 2, \ldots$ **do**
  observe y_t
  $(\hat{x}_{t|t}, P_{t|t}) \leftarrow \text{KFUpdate}(\hat{x}_{t|t-1}, P_{t|t-1}, f_{\text{enc}}(y_t))$
  $u_{t:t+H-1} \leftarrow$ solution of (48) using $z^*_{t+1:t+H}$
  apply u_t
  $(\hat{x}_{t+1|t}, P_{t+1|t}) \leftarrow \text{KFPredict}(\hat{x}_{t|t}, P_{t|t}, u_t)$
end for

The closed-loop procedure is summarized in Alg. 1.

F.2 Behavioral targets

Given the behavioral readout (W_z, b_z), the MPC solves the box-constrained quadratic program

$$\begin{aligned} \min_{u_{t:t+H-1}} \quad & \sum_{k=t}^{t+H-1} \Big[\big(W_z \hat{x}_{k+1|t} + b_z - z^*_{k+1}\big)^\top Q \big(W_z \hat{x}_{k+1|t} + b_z - z^*_{k+1}\big) + u_k^\top R u_k \Big] \\ \text{s.t.} \quad & \hat{x}_{k+1|t} = A\hat{x}_{k|t} + Bu_k, \\ & u_{\min} \le u_k \le u_{\max}, \qquad k = t, \ldots, t+H-1. \end{aligned} \tag{48}$$

For a constant behavioral target, $z^*_{k+1} = z^*$ throughout the planning horizon. The closed-loop procedure is summarized in Alg. 2.

F.3 Behavioral targets with the locally linear model

Given a nominal input sequence $\bar{u}_{t:t+H-1}$, the corresponding nominal trajectory is

$$\bar{x}_{t|t} = \hat{x}_{t|t}, \tag{49}$$

$$\bar{x}_{k+1|t} = A(\bar{x}_{k|t})\bar{x}_{k|t} + B(\bar{x}_{k|t})\bar{u}_k, \qquad k = t, \ldots, t+H-1. \tag{50}$$

The state-dependent matrices are then evaluated along this trajectory,

$$A_{k|t} = A(\bar{x}_{k|t}), \qquad B_{k|t} = B(\bar{x}_{k|t}), \tag{51}$$

and held fixed while solving the box-constrained quadratic program

$$\begin{aligned} \min_{u_{t:t+H-1}} \quad & \sum_{k=t}^{t+H-1} \Big[\big(W_z \hat{x}_{k+1|t} + b_z - z^*_{k+1}\big)^\top Q \big(W_z \hat{x}_{k+1|t} + b_z - z^*_{k+1}\big) + u_k^\top R u_k \Big] \\ \text{s.t.} \quad & \hat{x}_{k+1|t} = A_{k|t}\hat{x}_{k|t} + B_{k|t}u_k, \\ & u_{\min} \le u_k \le u_{\max}, \qquad k = t, \ldots, t+H-1. \end{aligned} \tag{52}$$

Algorithm 3 Control to a behavioral reference with the locally linear model

Require: learned model $(A(\cdot), B(\cdot), C(\cdot), \Sigma_x(\cdot), \Sigma_a(\cdot), f_{\text{enc}})$; behavioral readout (W_z, b_z); reference z^* or $\{z_t^*\}$; horizon H; weights Q, R; input box $[u_{\min}, u_{\max}]$; iterations N_{it}

$\hat{x}_{1|0} \leftarrow 0,\ P_{1|0} \leftarrow I,\ \bar{u}_{1:H} \leftarrow 0$
for $t = 1, 2, \ldots$ **do**
  observe y_t
  $(\hat{x}_{t|t}, P_{t|t}) \leftarrow \text{KFUpdate}(\hat{x}_{t|t-1}, P_{t|t-1}, f_{\text{enc}}(y_t))$
  for $i = 1, \ldots, N_{\text{it}}$ **do**
    $\bar{x}_{t|t} \leftarrow \hat{x}_{t|t}$
    for $k = t, \ldots, t + H - 1$ **do**
      $A_{k|t} \leftarrow A(\bar{x}_{k|t})$
      $B_{k|t} \leftarrow B(\bar{x}_{k|t})$
      $\bar{x}_{k+1|t} \leftarrow A_{k|t}\bar{x}_{k|t} + B_{k|t}\bar{u}_k$
    end for
    $\bar{u}_{t:t+H-1} \leftarrow$ solution of (52) using $z^*_{t+1:t+H}$
  end for
  apply $u_t = \bar{u}_t$
  $(\hat{x}_{t+1|t}, P_{t+1|t}) \leftarrow \text{KFPredict}(\hat{x}_{t|t}, P_{t|t}, u_t)$
  $\bar{u}_{t+1:t+H} \leftarrow (\bar{u}_{t+1}, \ldots, \bar{u}_{t+H-1}, \bar{u}_{t+H-1})$
end for

The optimized sequence becomes the nominal sequence for the next iteration. After N_{it} iterations, its first input is applied, and the remaining sequence is shifted to initialize the next control step.

G Experiment settings

G.1 Manifold plants

The plant state $x_t \in \mathbb{R}^{n_x}$ evolves linearly and is observed through the manifold embedding, $y_t = m(x_t) + s_t$ (11), with x restricted to the chart $[-\pi, \pi]$ (ring), $[-\pi, \pi]^2$ (torus) or $[0, 2\pi] \times [-\pi, \pi]$ (Swiss roll). Within the chart m is one-to-one, so the observation determines the state up to noise. Beyond the chart it is not: the ring and the torus are periodic, so y determines x only modulo 2π, and the Swiss roll continues into a second branch of the spiral. The unwrapped state on which the dynamics are linear is then not a function of the observation, and no continuous encoder can recover it. We therefore keep the experiments inside the chart: training episodes whose state leaves the chart at any step are rejected and resampled, and control targets are restricted to the chart. Rejection is not applied to the shear family, where too few episodes would remain; its training data therefore include episodes that leave the chart.

G.1.1 1D latent: ring manifold

We begin with the simplest non-trivial setting: a scalar latent state on a ring-like manifold. Let $x_t, u_t \in \mathbb{R}, y_t \in \mathbb{R}^3$, $A = e^{-\frac{1}{\tau}}$, $B = 1$ and $m(\cdot)$ be the manifold:

$$m(x) = \begin{bmatrix} \cos(x) \\ \sin(2x) \\ \sin(x) \end{bmatrix} \tag{53}$$

To establish ConFINE's ability to fit and control systems across the timescales relevant to neural data, we sweep τ from 5 to 200 samples, corresponding to eigenvalues $\lambda = e^{-1/\tau} \in [0.819, 0.995]$. This range covers the timescales reported across the literature: ECoG timescales span 10–50 ms across human cortex (Gao et al., 2020), and single-unit spike-count autocorrelation timescales span 50–350 ms across macaque sensory, parietal, and prefrontal areas (Murray et al., 2014). Recordings of these signals are sampled from 100 Hz, sufficient for clinical seizure localization (Davis et al., 2018), to 5 kHz for high-frequency oscillation analysis (Weiergräber et al., 2016); at a representative $f_s = 500$ Hz, our sweep $\tau \in [5, 200]$ samples maps to $\tau_c = \tau / f_s \in [10, 400]$ ms, fully containing

the biologically reported range. Timescales on either side can be brought into range by adjusting f_s: at $f_s = 100$ Hz the same sweep reaches $\tau_c = 2000$ ms, and at $f_s = 5$ kHz it resolves $\tau_c = 1$ ms.

G.1.2 2D LATENT: TORUS AND SWISS-ROLL MANIFOLDS

We next consider 2-D latents to access dynamics regimes that are inaccessible in 1-D: anisotropic decay, oscillation, and non-normal coupling. Throughout, $x_t, u_t \in \mathbb{R}^2$, $y_t \in \mathbb{R}^3$, $B = I_2$, and $m(\cdot)$ is either a toroidal manifold,

$$m\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} (r_1 + r_2\cos(x_1))\cos(x_2) \\ (r_1 + r_2\cos(x_1))\sin(x_2) \\ r_2\sin(x_1) \end{bmatrix} \tag{54}$$

with $r_1 = 4, r_2 = 1.5$, or a "Swiss roll",

$$m\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} \frac{x_1}{2}\cos(x_1) \\ x_2 \\ \frac{x_1}{2}\sin(x_1) \end{bmatrix} \tag{55}$$

We sweep A across three families targeting distinct, biologically motivated regimes.

Real modes. First, we vary independent decay time constants $\tau_1, \tau_2 \in [5, 200]$ along the two latent axes, with the biological calibration of this range identical to that of App. G.1.1:

$$A(\tau_1, \tau_2) = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \qquad \text{where } \lambda_i = e^{-\frac{1}{\tau_i}} \tag{56}$$

Oscillatory modes. Second, we sweep decay $\tau \in [5, 200]$ as before, jointly with oscillation frequency $\theta \in [\frac{\pi}{14}, \frac{\pi}{2}]$:

$$A(\tau, \theta) = e^{-\frac{1}{\tau}} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{57}$$

At $f_s = 500$ Hz, this range of θ corresponds to oscillations $f_{\text{osc}} = \theta f_s/(2\pi)$ from 18 to 125 Hz, spanning the beta (10–30 Hz; Buzsáki & Draguhn, 2004, Fig. 1B) and gamma (30–90 Hz; Buzsáki et al., 2012) bands; lower bands are reached at lower sampling rates (3.6–25 Hz at $f_s = 100$ Hz), and at $f_s = 2$ kHz the range extends to 500 Hz, capturing high gamma (Gaona et al., 2011), hippocampal ripples (Ylinen et al., 1995), and fast ripples (Bragin et al., 1999; 2010).

Non-normal modes. Third, we consider non-normal systems, which produce transient growth of perturbations along non-orthogonal eigendirections, before decaying asymptotically. Such systems are ubiquitous in natural and engineered networks (Asllani et al., 2018) and intrinsic to cortical circuits, where Dale's law produces a hidden feedforward structure supporting bounded transient amplification (Murphy & Miller, 2009; Hennequin et al., 2012; Ganguli et al., 2008). We sweep the geometric-mean decay time $\bar{\tau} \in [10, 50]$ (at ratio $\rho = 4$, yielding $\tau_1 \in [20, 100]$ and $\tau_2 \in [5, 25]$), jointly with shear $s \in [0.1, 3]$, which controls non-normality:

$$A(\bar{\tau}, s) = \begin{bmatrix} \lambda_1 & -s \\ 0 & \lambda_2 \end{bmatrix} \qquad \text{where } \tau_1 = \bar{\tau}\sqrt{\rho}, \quad \tau_2 = \frac{\bar{\tau}}{\sqrt{\rho}} \tag{58}$$

The upper bound $s = 3$, combined with the chosen range of $\bar{\tau}$, covers the range of non-normality metrics reported for neural circuits: the normalized Henrici ratio (Asllani et al., 2018) $\hat{d}_F(A) = \frac{\sqrt{||A||_F^2 - \sum_i |\lambda_i|^2}}{||A||_F}$ ranges up to 0.92 and peak transient amplification $G_{\max}(A) = \max_k \|A^k\|_2$ up to ≈ 48 (Fig. A15). These bounds include values reported in the literature: $\hat{d}_F \approx 0.4$–0.6 in linearized connectome dynamics (Nazerian et al., 2026, Fig. 4) and up to ≈ 0.92 in stability-optimized networks obeying Dale's law (Christodoulou et al., 2022, Fig. 7F); and transient amplification of ≈ 1.1–$1.3\times$ in mouse auditory cortex (Bondanelli et al., 2021, Fig. 3G), ≈ 3–$10\times$ in simulated balanced networks (Christodoulou et al., 2022, Fig. 7E; Murphy & Miller, 2009, Fig. 2), and $\approx 25\times$ in evoked energy (Hennequin et al., 2014).

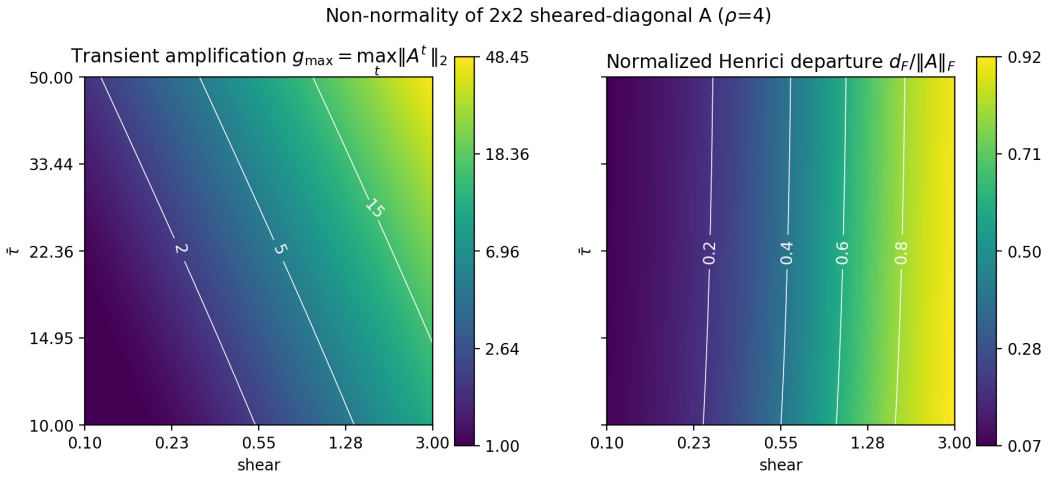

Figure A15: Non-normality of the shear family over the sweep: peak transient amplification $g_{\max} = \max_t \|A^t\|_2$ (left, log color scale) and normalized Henrici departure $d_F/\|A\|_F$ (right) as functions of the shear s and the mean time constant $\bar{\tau}$ ($\rho = 4$).

G.1.3 Experimental protocol.

Each family is a grid of plants over its two parameters, 7 values per axis (49 plants; 30 values of τ on the ring), log-spaced except θ, which is linearly spaced. The latent state has process noise of standard deviation 0.045, and y has observation noise of standard deviation 0.1. For each plant we train ConFINE, its ablation without KL and AE+LDM with 5 seeds each, and fit LDM once (subspace identification is deterministic), on 2^{14} episodes of 50 steps driven by inputs $u_t \sim \mathcal{U}(-0.5, 0.5)^{n_u}$ drawn independently at every step. The control targets are a grid of 20 points per latent dimension over the plant's chart (400 targets, 20 on the ring), each mapped to the manifold, $y^* = m(x^*)$; the input box for control is the training box, enlarged per plant where necessary to the smallest box in which every target can be held at equilibrium. Every episode starts at the center of the chart and control switches on at step 5 of 75. Since the task is to reach and hold the target, the cost of a trained model is the Euclidean distance between the measured observation y_t and y^* averaged over the last 10 steps, over the targets, and over 10 noise realizations per target. A plant's cost is the median over its seeds, and the score of a model is the mean over all plants; no plant is excluded, so every model is scored on the same plants. Two reference levels bracket the reported costs: the *oracle* runs the same MPC with the true plant and true state and gives the reference floor, and *chance* is the average target-to-target distance, the cost of settling on a random target.

G.2 Reach RNN

Plant. The network has 32 rectified-linear units, a 2-d input u_t, a 2-d behavioral output z_t (hand position) and a 32-d observation y_t:

$$h_{t+1} = \phi(W_h h_t + W_u u_t) + q_t, \qquad q_t \sim \mathcal{N}(0, \sigma_p^2 I) \tag{59}$$

$$y_t = g_\theta(h_t) + r_t, \qquad r_t \sim \mathcal{N}(0, \sigma_o^2 I) \tag{60}$$

$$z_t = W_z h_t \tag{61}$$

where g_θ is a fixed random two-layer tanh network with 32 output units, and σ_p and σ_o are the process and observation noise standard deviations. The recurrent weights W_h are initialized with gain 1.2 and, together with W_u and W_z, trained on a center-out reach: for a persistent input $u_t = u^* \sim \mathcal{U}([-1, 1]^2)$ held for 50 steps, the output is trained to reach $z = u^*$ along an exponential profile $z_t^{\text{ref}} = z_0 + (u^* - z_0)(1 - e^{-t/12})$ (4096 targets, 601 epochs). The network is then frozen and noise is added; we use three variants, A ($\sigma_p = 0.01$, episodes starting with $z_0 \in [-0.25, 0.25]^2$), B ($\sigma_p = 0.01$, $z_0 \in [-1, 1]^2$) and C ($\sigma_p = 0.03$, $z_0 \in [-1, 1]^2$), all with $\sigma_o = 0.3$.

Data and models. Models are trained on 2^{12} episodes of 100 steps driven by random inputs $u_t \sim \mathcal{U}([-1,1]^2)$ held for 1 or 4 steps, with initial states drawn from the same box the control episodes start in; the model sees only (y, u), never the behavior. We compare ConFINE, its ablation without KL, AE+LDM (8 training seeds each) and LDM (a single fit, since subspace identification is deterministic). After training, the behavior readout is fitted by least squares on the model's filtered states (Eq. 9 with $\mathcal{K}_z = \{0\}$), so behavior is used only to fit the readout, not to train the model.

Control task. The controller drives the behavior to one of 8 targets, $(\pm 0.5, \pm 0.5)$ and the four axis points at ± 0.5, from a random initial state, with control switching on at step 5 of a 55-step episode; each model is evaluated on 25 episodes per target with plant-noise realizations shared across models. The reported cost is the planner's objective evaluated on the plant's behavior, $\sum_t \|z_t - z^*\|^2 + R \sum_t \|u_t\|^2$, summed from the onset of control, at two input penalties $R \in \{10^{-2}, 10^{-3}\}$. The oracle is an MPC with the same horizon and cost that plans through the true network from its true state; "do nothing" applies $u = 0$. Reported values are mean $\pm$ s.e. over all evaluations pooled (training seeds $\times$ 25 episodes $\times$ 8 targets).

G.3 Grasping-circuit model

Plant. We use the plant of Bryan et al. (2023): the modular recurrent network of Michaels et al. (2020), with 100 units in each of the anterior intraparietal area (AIP), the hand area of ventral premotor cortex (F5), and the hand area of primary motor cortex (M1), trained to reproduce a monkey's 50 muscle velocities during a delayed reach-to-grasp task and shown to resemble neural recordings from the three areas. The behavior z_t is this muscle-velocity output, a linear readout of M1. Each area is observed through 20 channels, each a Gaussian-weighted average of the activity of the area's units, and stimulation enters through 16 channels bounded to $[-1, 1]$. Following Bryan et al. (2023), we simulate four lesions: *M1* silences half of M1 and stimulates the other half; *AIP* silences half of AIP and stimulates M1; *connection* cuts the projections between M1 and F5, disconnecting M1, and stimulates M1; *AIP+F5+conn* cuts the projections between AIP and F5, disconnecting AIP, and stimulates half of F5 and half of AIP. M1 is not recorded in the M1 and AIP lesions (40 channels); all three areas are recorded in the other two (60 channels).

Control task. The controlled epoch is the 59 steps after the go cue of each trial, and the reference z_t^* is the muscle-velocity trajectory recorded on that trial. Because the network is deterministic, every repetition of a trial would otherwise reach the go cue in the same state, leaving nothing to estimate; we therefore drive the pre-go period with random stimulation that the model does not observe, which spreads the state at the go cue (per-unit standard deviation 0.032), and start the filter from its prior at the go cue. The healthy, lesioned and controlled rollouts of a trial receive the same warm-up, so the three losses below start from the same state. We report percent recovery, $100\,(\ell_{\text{lesion}} - \ell_{\text{control}})/(\ell_{\text{lesion}} - \ell_{\text{healthy}})$, where ℓ_{control}, ℓ_{lesion} and ℓ_{healthy} are the mean squared errors between the produced and the recorded kinematics over the 59 steps for, respectively, the lesioned circuit driven by the controller, the lesioned circuit without stimulation, and the healthy circuit without stimulation, all on the same trials. Recovery is thus 0% when the controlled circuit does no better than the unstimulated lesioned circuit and 100% when it matches the healthy circuit; negative values mean that stimulation made the movement worse than no stimulation, and values above 100% are possible because the reference is the recorded movement, which the healthy circuit reproduces only approximately. Values are mean $\pm$ s.e. over 4 training seeds $\times$ 100 held-out trials.

Data and models. The dataset contains 502 recorded trials, each a reach with its own visual input and recorded kinematics, which we split at random into 402 identification trials and 100 held-out trials. The training set consists of 2^{14} episodes, each generated by running the lesioned circuit on a randomly drawn identification trial with random stimulation: during the pre-go warm-up, and during the 59-step reach epoch that forms the training sequence, where stimulation is drawn uniformly from $[-1, 1]^{16}$ and held for 1, 5, 15, 30 or 59 steps. There is no closed-loop interaction during training. Control is then evaluated on the 100 held-out trials, whose recorded kinematics serve as the references. We compare ConFINE and LL-ConFINE, each with and without the KL term, with LDM and AE+LDM (App. E.2), under two protocols: *unsupervised*, in which the behavior readout is fitted after training via (9), and *supervised*, in which it is trained jointly.

Table 4: Model and training settings per experiment. Encoder widths are hidden-layer sizes; the decoder mirrors them. $N \times T$: number and length of training sequences. Step weights: weight of the $k = 0$ term versus each $k \geq 1$ term. Behavior readout: fitted post hoc via (9) or trained jointly with the model through $\mathcal{L}_z$ (supervised protocol); $\mathcal{K}_z$: its horizons in either case; λ_z: weight of $\mathcal{L}_z$.

	Manifolds	Reach RNN	Grasping circuit
n_y / n_u	3 / 1 (ring), 3 / 2 (torus, Swiss roll)	32 / 2	40 or 60 / 16
n_a / n_x	1 / 1 (ring), 2 / 2 (torus, Swiss roll)	32 / 32	32 / 16
Encoder	$[32, 32, 32]$, ReLU	$[256, 256]$, ELU	$[128, 128]$, ELU
$N \times T$	$2^{14} \times 50$	$2^{12} \times 100$	$2^{14} \times 59$
Chunk length	32	48	32
Batch size	64	64	128
Epochs	13 (3 burn-in)	50	100
Learning rate	3×10^{-2}	10^{-2}	3×10^{-3}
Weight decay	0	0	10^{-6}
Step weights ($k = 0 : k \geq 1$)	1 : 1	1 : 1	8 : 1
KL weight / free nats α	0.1 / 1	10^{-2} / 5	10^{-2} / 3
Behavior readout	–	post hoc	post hoc or joint
Readout horizons $\mathcal{K}_z$	–	$\{0\}$	$\{0, \ldots, 8\}$
Joint readout weight λ_z	–	–	0.03

G.4 CONTROL

Planner. All controllers are receding-horizon MPC with horizon $H = 8$, each step solving a box-constrained quadratic program. On the manifolds the target pair (x^*, u^*) is the least-squares solution of the stacked system $Cx = \tilde{a}^*$, $(I - A)x - Bu = 0$ with u restricted to the input box, i.e. $\lambda_{\text{eq}} = 1$, and the stage cost is $(x - x^*)^\top Q(x - x^*) + (u - u^*)^\top R(u - u^*)$. On the reach RNN and the grasping-circuit model the stage cost is $||W_z\hat{x} + b_z - z^*||^2 + u^\top Ru$, with z standardized per channel on the grasping-circuit model. The locally linear models use the iterative planner with $N_{\text{it}} = 5$ iterations per control step, warm-started from the previous step's plan shifted by one. Table 5 lists the remaining settings.

Table 5: Control settings per experiment.

	Manifolds	Reach RNN	Grasping circuit
Target	y^* (observation)	z^* (behavior, constant)	z^*_k (behavior, reference trajectory)
Q	I	I	I
R	$10^{-3}I$	$10^{-2}I, 10^{-3}I$	$10^{-6}I$
u^*	from the fixed point	0	0
Input box	training box†	$[-1, 1]^2$	$[-1, 1]^{16}$
Episode length / control on	75 / 5	55 / 5	59 / 1
Targets $\times$ episodes	20^{n_x} grid $\times$ 10	8 $\times$ 25	100 trials

†enlarged, where necessary, to the smallest box in which every target of the grid is holdable.